F 9

```
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
LLL                 IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRRRRRRRRRRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR         RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR         RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR         RRR
LLL                    III        NNNNNN      NNN  KKK      KKK     EEE              RRR         RRR
LLL                    III        NNNNNN      NNN  KKK      KKK     EEE              RRR         RRR
LLL                    III        NNNNNN      NNN  KKK      KKK     EEE              RRR         RRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEEE     RRRRRRRRRRRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEEE     RRRRRRRRRRRR
LLL                    III        NNN   NNN   NNN  KKKKKKKKK        EEEEEEEEEEEE     RRRRRRRRRRRR
LLL                    III        NNN      NNNNNN  KKK      KKK     EEE              RRR   RRR
LLL                    III        NNN      NNNNNN  KKK      KKK     EEE              RRR   RRR
LLL                    III        NNN      NNNNNN  KKK      KKK     EEE              RRR   RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR      RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR      RRR
LLL                    III        NNN         NNN  KKK         KKK  EEE              RRR      RRR
LLLLLLLLLLLLLLL     IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRR         RRR
LLLLLLLLLLLLLLL     IIIIIIIII     NNN         NNN  KKK         KKK  EEEEEEEEEEEEEEE  RRR         RRR
```

```
**FILE**ID**LNKIMGOUT

LL          NN      NN  KK      KK    IIIIII    MM      MM    GGGGGGGG    OOOOOO    UU      UU  TTTTTTTTTT
LL          NN      NN  KK      KK    IIIIII    MM      MM    GGGGGGGG    OOOOOO    UU      UU  TTTTTTTTTT
LL          NN      NN  KK      KK      II      MMMM  MMMM  GG          OO      OO  UU      UU      TT
LL          NN      NN  KK      KK      II      MMMM  MMMM  GG          OO      OO  UU      UU      TT
LL          NNNN    NN  KK    KK        II      MM  MM  MM  GG          OO      OO  UU      UU      TT
LL          NNNN    NN  KK    KK        II      MM  MM  MM  GG          OO      OO  UU      UU      TT
LL          NN  NN  NN  KKKKKK          II      MM      MM  GG          OO      OO  UU      UU      TT
LL          NN  NN  NN  KKKKKK          II      MM      MM  GG          OO      OO  UU      UU      TT
LL          NN    NNNN  KK    KK        II      MM      MM  GG  GGGGGG  OO      OO  UU      UU      TT
LL          NN    NNNN  KK    KK        II      MM      MM  GG  GGGGGG  OO      OO  UU      UU      TT
LL          NN      NN  KK      KK      II      MM      MM  GG      GG  OO      OO  UU      UU      TT          ....
LL          NN      NN  KK      KK      II      MM      MM  GG      GG  OO      OO  UU      UU      TT          ....
LLLLLLLLLL  NN      NN  KK      KK    IIIIII    MM      MM    GGGGGG      OOOOOO    UUUUUUUUUU      TT          ....
LLLLLLLLLL  NN      NN  KK      KK    IIIIII    MM      MM    GGGGGG      OOOOOO    UUUUUUUUUU      TT          ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0 module lnk_imgout
0               (ident = 'V04-000'
0               ,addressing_mode
0                       (external    = general
0                       ,nonexternal = long_relative
0                       )
0               ) =
1 begin
1 !
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1 !
1 !
1 !
1 !
1 !++
1 !
1 ! MODULE:        LNK_IMGOUT
1 !
1 ! FACILITY:      LINKER
1 !
1 ! ABSTRACT:      Control production of image file
1 !
1 ! HISTORY:
1 !
1 !    VERSION:    X01.00
1 !
1 !    AUTHOR:     T.J. PORTER 06-APR-77
1 !
1 !    MODIFIED BY:
1 !
1 !       V03-033 ADE0011         Alan D. Eldridge        7-Aug-1984
1 !               Add support for LNK$GL_UNSUPPORTED.  When the low bit
1 !               is set and we're building a shareable image, then don't convert
1 !               demand zero pages to copy on reference.  This is done to get
```

```
1 !                     around a bug in the image activator that does not know how
1 !                     to deal with shareable demand zero sections.
1 !
1 !                     Change number of useable bytes in all other blocks but the first
1 !                     in the image-header to 508 (was 512).  This is done to avoid a
1 !                     bug in the image-activator.
1 !
1 !                     Exit LNK$WRTIMGBYTS immediately if the 'repeat-count' arguement
1 !                     is zero.
1 !
1 !             V03-033 ADE0010         Alan D. Eldridge        30-Jul-1984
1 !                     Use error returned by LNK$REQUESTMEM rather than assuming
1 !                     LIN$_MEMFUL.
1 !
1 !                     Change the number of useable bytes in the first block of the
1 !                     image header to 508 (from 510).  The last word of the first
1 !                     block has a special meaning, and the second to the last word
1 !                     is unusable due to a bug in the image-activator.
1 !
1 !             V03-032 JWT0189         Jim Teague             25-Jul-1984
1 !                     Re-implement demand zero pages in shareable images.
1 !
1 !             V03-031 ADE0009         Alan D. Eldridge        5-Jul-1984
1 !                     Add in fixup section page count when figuring the number of
1 !                     blocks in the image portion of the image file.
1 !
1 !                     Disallow demand zero pages in shareable images until side
1 !                     effects (no symbol table at end of image) have been fixed.
1 !
1 !             V03-030 ADE0008         Alan D. Eldridge        22-Jun-1984
1 !                     Adhere to Grammer Rules for output file spec's as defined
1 !                     in the Command Language User's Guide.
1 !
1 !                     Allow demand zero pages in shareable images.
1 !
1 !             V03-029 ADE0007         Alan D. Eldridge        19-May-1984
1 !                     More fixes to V03-026.
1 !
1 !             V03-028 ADE0006         Alan D. Eldridge        18-May-1984
1 !                     Don't initialize header if there is none -- fix to
1 !                     update V03-026.
1 !
1 !             V03-027 ADE0005         Alan D. Eldridge        16-May-1984
1 !                     Allocate maximum sized block to store list of shared,
1 !                     local, and based ISD's.
1 !
1 !             V03-026 ADE0004         Alan D. Eldridge         8-May-1984
1 !                     Fix image header construction when image header is longer
1 !                     than one block.
1 !
1 !             V03-025 JWT0180         Jim Teague             26-Apr-1984
1 !                     Fill in SHL$B_SHL_SIZE field in first shareable
1 !                     image list entry.
1 !
1 !             V03-024 ADE0003         Alan D. Eldridge        26-Mar-1984
1 !                     Default .EXE device and directory based on whether the /EXE
1 !                     in the command was a command or input parameter qualifier.
```

```
1 !
1 !            V03-023 ADE0002         Alan D. Eldridge        25-Mar-1984
1 !                    Add 5th entry to transfer vector array for initializing shared
1 !                    images.
1 !
1 !            V03-022 ADE0001         Alan D. Eldridge         3-Mar-1984
1 !                    Make default GSMATCH is created with high bit clear since
1 !                    the high bit set now means to skip the date checks for
1 !                    shareable images and the corresponding library entry.
1 !
1 !                    Fix bug which limited shareable image name lengths
1 !                    to 30 instead of 31 characters.
1 !
1 !            V03-021 JWT0153         Jim Teague               8-Feb-1984
1 !                    Remove restriction imposed by JWT0131.  Use expanded
1 !                    image name field.  Now 40 bytes: 1 for count, 39 for
1 !                    file name.
1 !
1 !            V03-020 JWT0131         Jim Teague              18-Aug-1983
1 !                    FT1 restriction on 11-character shareable image
1 !                    names.
1 !
1 !            V03-019 JWT0113         Jim Teague              20-Apr-1983
1 !                    Do $getjpi calls to get number of open files left.
1 !
1 !            V03-018 JWT0088         Jim Teague              12-Jan-1983
1 !                    Filter 0-length psects out of DMT.
1 !
1 !            V03-017 JWT0078         Jim Teague              15-Dec-1982
1 !                    Change implementation of DMT.
1 !
1 !            V03-016 JWT0077         Jim Teague              13-Dec-1982
1 !                    Slight ECO to JWT0074.
1 !
1 !            V03-015 JWT0074         Jim Teague              09-Dec-1982
1 !                    Finalize Debugger DMT.
1 !
1 !            V03-014 JWT0055         Jim Teague              15-Sep-1982
1 !                    Make uninitialized copy-on-ref image sections demand-zero
1 !                    even if smaller than DZRO_MIN.
1 !
1 !            V03-013 JWT0050         Jim Teague              11-Aug-1982
1 !                    Place cli image flag in final word of first image header
1 !                    block if IMAGE_TYPE=CLI specified in options file.
1 !
1 !            V03-012 JWT0044         Jim Teague              30-Jul-1982
1 !                    Open file performance boost.
1 !
1 !            V03-011 JWT0038         Jim Teague              23-Jun-1982
1 !                    Clean up INFO#212 error.  Also change $EXPREG calls to
1 !                    LNK$REQUESTMEM which will make the calls with better
1 !                    error hanndling.
1 !
1 !--
```

```
;   170       0169  1 !
;   171       0170  1 !++
;   172       0171  1 !
;   173       0172  1 ! FUNCTIONAL DESCRIPTION:
;   174       0173  1 !
;   175       0174  1 !  This module contains all routines associated with image output.
;   176       0175  1 !  It contains the following externally called routines:
;   177       0176  1 !
;   178       0177  1 !        LNK$IMGINIT       No arguments
;   179       0178  1 !
;   180       0179  1 !        LNK$WRTIMGHDR     No arguments
;   181       0180  1 !
;   182       0181  1 !                          Processes the singly linked list of image section
;   183       0182  1 !                          descriptors, building the image header with all its
;   184       0183  1 !                          ISD'S.
;   185       0184  1 !
;   186       0185  1 !        LNK$WRTIMGBYTS (BYTECOUNT,BUFFRADR,REPTCOUNT)
;   187       0186  1 !
;   188       0187  1 !                          Writes "BYTECOUNT" bytes from "BUFFRADR" "REPTCOUNT"
;   189       0188  1 !                          times and updates the linker's location counter by the
;   190       0189  1 !                          total number of bytes.  If REPTCOUNT or BYTECOUNT is
;   191       0190  1 !                          zero, then exit immediately.
;   192       0191  1 !
;   193       0192  1 !                          Attempts to write a byte stream, any part of which falls
;   194       0193  1 !                          outside the space allocated to the image produces an
;   195       0194  1 !                          error and the value false is returned.
;   196       0195  1 !
;   197       0196  1 !        LNK$FLUSHIMG      No arguments
;   198       0197  1 !
;   199       0198  1 !                          Writes every block of image, ensuring that those not
;   200       0199  1 !                          touched heretofore are zero filled.
;   201       0200  1 !
;   202       0201  1 !        LNK$CLOSIMGFIL    No arguments
;   203       0202  1 !
;   204       0203  1 !                          Closes the image file.
;   205       0204  1 !
;   206       0205  1 !        ANY OF THE ABOVE ERRORS TURNS OFF IMAGE PRODUCTION
;   207       0206  1 !        (BIT LNK$S_IMAGE IN LNK$GL_CTLMSK).
;   208       0207  1 !
;   209       0208  1 !
;   210       0209  1 !--
;   211       0210  1 !
;   212       0211  1 library
;   213       0212  1         'LIBL32';                                         ! SYSTEM USER DATA STRUCTURES
;   214       0213  1 require
;   215       0214  1         'PREFIX';
;   216       0329  1 library
;   217       0330  1         'DATBAS';                                         ! INTERNAL DATABASE
;   218       0331  1 require
;   219       0332  1         'ISGENC';                                         ! IMAGE SECTION FORMATS
;   220       0716  1
;   221       0717  1 forward routine
;   222       0718  1         lnk$closimgfil   : novalue,                       ! CLOSE ROUTINE
;   223       0719  1         inchdrsize       : novalue,                       ! ROUTINE THAT ACCUMULATES HEADER SIZE
;   224       0720  1         isdtoheader      : novalue,                       ! COPY AN ISD INTO THE HEADER
;   225       0721  1         writeisects,                                      ! WRITE ALL ISECTS TO IMAGE
;   226       0722  1         lnk$getfixsize,                                   ! COMPUTE SIZE OF FIXUP SECTION
;  227        0723  1               storespcgfix,                                   ! STORE SPECIAL G^ FIXUP DATA
;  228        0724  1               lnk$createfix,                                  ! CREATE AND INIT FIXUP SECTION
;  229        0725  1               readsection      : novalue,                     ! READ GLOBAL SECTIONS FROM SHAREABLE IMAGES
;  230        0726  1               writesection;                                   ! WRITES A SECTION OF AN IMAGE
;  231        0727  1
;  232        0728  1       external routine
;  233        0729  1               lnk$closefile    : novalue,                     ! CLOSE A FILE
;  234        0730  1               lnk$requestmem,                                 ! MAKE A $EXPREG REQUEST
;  235        0731  1               lib$traverse_tree,                              ! TRAVERSE A BINARY TREE
;  236        0732  1               lnk$filnamdsc,                                  ! GET FILE NAME FROM FAB
;  237        0733  1               lnk$alloblk      : novalue,                     ! ALLOCATE DYNAMIC MEMORY
;  238        0734  1               lnk$dealblk      : novalue,                     ! DEALLOCATE DYNAMIC MEMORY
;  239        0735  1               lnk$allovirmem,                                 ! ALLOCATE VIRTUAL MEMORY
;  240        0736  1               lnk$crefixisd,                                  ! CREATE FIXUP ISD
;  241        0737  1               lnk$sortisects   : novalue,                     ! SORT IMAGE SECTION LIST
;  242        0738  1               lnk$search;                                     ! SYMBOL TABLE SEARCH
;  243        0739  1
;  244        0740  1       external literal
;  245        0741  1               lin$_closein,                                   ! ERROR CLOSING INPUT FILE
;  246        0742  1               lin$_closeout,                                  ! ERROR CLOSING OUTPUT FILE
;  247        0743  1               lin$_confixmem,                                 ! BASED IMAGE HAPPENED ON TOP OF FIXUP SECTION
;  248        0744  1               lin$_dbgtfr,                                    ! NO DEBUGGER TRANSFER ADDRESS
;  249        0745  1               lin$_getptierr,                                 ! ERROR RETURNED FROM SYS$GETPTI
;  250        0746  1               lin$_memful,                                    ! FAILED TO MAP THE IMAGE
;  251        0747  1               lin$_noimgfil,                                  ! NO IMAGE FILE CREATED
;  252        0748  1               lin$_openin,                                    ! ERROR OPENING INPUT FILE
;  253        0749  1               lin$_openout,                                   ! ERROR OPENING OUTPUT FILE
;  254        0750  1               lin$_outsimg,                                   ! ATTEMPT TO STORE OUTSIDE IMAGE
;  255        0751  1               lin$_outsimgp,                                  ! DITTO, BUT WITH A PSECT NAME
;  256        0752  1               lin$_readerr,                                   ! READ FAILURE ON SHAREABLE IMAGE
;  257        0753  1               lin$_tfrsys,                                    ! TRANSFER ADDRESS IN SYSTEM IMAGE
;  258        0754  1               lin$_usrtfr,                                    ! NO USER TRANSFER ADDRESS IN EXE IMAGE
;  259        0755  1               lin$_writeerr,                                  ! WRITE ERROR CODE
;  260        0756  1               lnk$K_max_filename_length ;                     ! MAXIMUM FILENAME LENGTH
;  261        0757  1
;  262        0758  1       external
;  263        0759  1               lnk$gl_unsupported      : bitvector [],         ! Bit mask of requested "unsupported" options
;  264        0760  1               lnk$gl_filesleft,
;  265        0761  1               lnk$gt_jpilst,
;  266        0762  1               lnk$gl_dmtbuffer        : vector [2],           ! BUFFER FOR DMT
;  267        0763  1               lnk$gl_dmtbytes,                                ! SIZE OF DMT IN BYTES
;  268        0764  1               lnk$gl_minva,                                   ! LOWEST ADDRESS VA ALLOCATED BY VMCTRL
;  269        0765  1               lnk$gl_lastgadr,                                ! LAST SPECIAL G^ FIXUP VA
;  270        0766  1               lnk$gl_spcgfix,                                 ! NUMBER OF SPECIAL G^ FIXUPS
;  271        0767  1               lnk$gl_spcgimgs,                                ! NUMBER OF SHR IMGS REFERENCED BY
;  272        0768  1                                                               ! SPECIAL G^ FIXUPS
;  273        0769  1               lnk$gl_adrefs,                                  ! NUMBER OF .ADDRESS FIXUP REFERENCES
;  274        0770  1               lnk$gl_shradr,                                  ! NUMBER OF SHR IMAGES REF'D BY .ADDRESS
;  275        0771  1               lnk$gl_shrimgs,                                 ! NUMBER OF PIC SHAREABLE IMAGES
;  276        0772  1               lnk$gl_shrsyms,                                 ! NUMBER OF SYMBOLS IN THOSE IMAGEES
;  277        0773  1               lnk$gl_shlextra,                                ! SHL_EXTRA REQUEST
;  278        0774  1               lnk$gl_fixisd           : ref block [,byte],    ! ISD OF FIXUP SECTION
;  279        0775  1               lnk$gl_lowclu           : ref block [,byte],    ! ADDR OF LOWEST VA NON-SHR IMG CLUSTER
;  280        0776  1               lnk$aw_version          : block [,byte],        ! VERSION NUMBER
;  281        0777  1               lnk$gt_sysver           : block [,byte],        ! "SYS$K_VERSION" ASCIC STRING
;  282        0778  1               lnk$gl_curomd           : ref block [,byte],    ! POINTER TO CURRENT OBJ MODULE DESCRIPTOR
;  283        0779  1               lnk$gl_curfil           : ref block [,byte],    ! POINTER TO CURRENT FILE FDB
;  284      0780  1        lnk$gl_curpsc          : ref block [,byte],    ! POINTER TO CURRENT PSECT DESCRIPTOR
;  285      0781  1        lnk$gl_inrelnam,                               ! POINTER TO 1ST INPUT FILE NAM BLOCK
;  286      0782  1        lnk$gl_relnam_exe,                             ! POINTER TO .EXE RELATED NAME
;  287      0783  1        lnk$gb_locnov_exe      : byte,                 ! LOW BIT SET IF /EXE WAS 'LOCAL' WITH NO VALUE
;  288      0784  1        lnk$gb_pass            : byte,                 ! PASS NUMBER
;  289      0785  1        lnk$gl_ctlmsk          : block [,byte],        ! CONTROL FLAGS
;  290      0786  1        lnk$gl_clulst,                                 ! CLUSTER LISTHEADE
;  291      0787  1        lnk$gl_curclu          : ref block [,byte],    ! CURRENT CLUSTER
;  292      0788  1        lnk$gl_lstclstr        : ref block [,byte],    ! POINTER TO HIGHEST ADDRESSED CLUSTER
;  293      0789  1        lnk$gl_defclu          : block [,byte],        ! DEFAULT CLUSTER BLOCK
;  294      0790  1        lnk$gl_imgfil          : ref block [,byte],    ! POINTER TO IMAGE DESCRIPTOR BLOCK
;  295      0791  1        lnk$gl_dbgestim,                               ! ESTIMATE OF DST BYTES ON PASS 1
;  296      0792  1        lnk$gl_dbgtfps,                                ! P-SECT DESCRIPTOR FOR DEBUGGER
;  297      0793  1                                                       ! TRANSFER ADDRESS
;  298      0794  1        lnk$gl_dbgtfr,                                 ! DEBUGGER TRANSFER ADDRESS
;  299      0795  1        lnk$gw_gstrecs         : word,                 ! NUMBER RECORDS WRITTEN TO GST OF IMAGE
;  300      0796  1        lnk$gl_tfrpsc,                                 ! USER TRANSFER ADDRESS P-SECT
;  301      0797  1        lnk$gl_tfradr,                                 ! USER TRANSFER ADDRESS
;  302      0798  1        lnk$gq_alostim         : vector,               ! TIME OF MEMORY ALLOCATION
;  303      0799  1        lnk$gq_privs           : vector [2,long],      ! VECTOR OF IMAGE PRIVILEGES
;  304      0800  1        lnk$gt_imgnam          : vector [,byte],       ! IMAGE NAME AREA
;  305      0801  1        lnk$gt_imgid           : vector [,byte],       ! IMAGE IDENT AREA
;  306      0802  1        lnk$gw_chans           : word,                 ! NUMBER OF I/O CHANNELS
;  307      0803  1        lnk$gw_dzromin         : word,                 ! NUMBER OF PAGES REQUIRED IN I-SECT BEFORE
;  308      0804  1                                                       ! COMPRESSION
;  309      0805  1        lnk$gw_ioseg           : word,                 ! NUMBER OF PAGES OF IMAGE I/O SEGMENT
;  310      0806  1        lnk$gw_misects         : word,                 ! MAXIMUM NUMBER OF IMAGE SECTIONS
;  311      0807  1        lnk$gw_nisects         : word,                 ! ACTUAL NUMBER OF IMAGE SECTIONS
;  312      0808  1        lnk$gw_shrisc ts       : word,                 ! NUMBER OF ISECTS FROM PIC SHAREABLE IMAGES
;  313      0809  1        lnk$gw_nsymbols        : word,                 ! NUMBER OF GLOBAL SYMBOLS
;  314      0810  1        lnk$gw_lsymbols        : word,                 ! NUMBER OF GLOBAL SYMBOLS IN THIS IMAGE
;  315      0811  1                                                       !   (NOT IN SHR IMAGE)
;  316      0812  1        lnk$gl_gsbufdsc        : vector [3] ;          ! DESCRIPTOR FOR COPYING GLOBAL SECTIONS
;  317      0813  1
;  318      0814  1 global
;  319      0815  1        lnk$al_imgrab          : $rab (rac=seq),       ! IMAGE FILE RECORD ACCESS BLOCK.
;  320      0816  1        lnk$gl_imghdr          : ref block [,byte],    ! POINTER TO IMAGE HEADER
;  321      0817  1        lnk$gl_curisd          : ref block [,byte],    ! CURRENT ISD ADDRESS
;  322      0818  1        lnk$gl_imgtop,                                 ! MAPPED ADDRESS AT TOP + 1
;  323      0819  1        lnk$gl_locn,                                   ! CURRENT LOCATION COUNTER
;  324      0820  1        lnk$gl_dstlocn,                                ! NEXT LOCATION IN DST
;  325      0821  1        lnk$gl_dstend,                                 ! END OF DST WRITTEN
;  326      0822  1        lnk$gl_rprotisd,                               ! NUMBER OF SECTIONS TO REPROTECT
;  327      0823  1        lnk$gl_shrclstrs,                              ! NUMBER OF SHAREABLE IMAGE CLUSTERS
;  328      0824  1                                                       !   (PIC AND NON-PIC)
;  329      0825  1        lnk$gw_nclusters       : word,                 ! NUMBER OF CLUSTERS CREATED
;  330      0826  1        lnk$gw_dstvbn          : word,                 ! VBN AT WHICH DST BEGINS
;  331      0827  1        lnk$gw_hdrblks         : word,                 ! BLOCKS IN IMAGE HEADER
;  332      0828  1        lnk$gw_imgblks         : word,                 ! BLOCKS OF IMAGE
;  333      0829  1        lnk$gw_imgifi          : word,                 ! IFI OF IMAGE FILE
;  334      0830  1        lnk$gw_dstblks         : word,                 ! NUMBER OF BLOCKS OF DST
;  335      0831  1        lnk$gw_gstvbn          : word,                 ! VBN OF GLOBAL SYMBOL TABLE
;  336      0832  1        lnk$gl_matchid,                                ! MATCH CONTROL ID FOR THIS SHAREABLE IMAGE
;  337      0833  1        lnk$gb_matchctl        : byte initial (%x'FF') ; ! MATCH CONTROL FLAGS
;  338      0834  1
;  339      0835  1 own
;  340      0836  1        dmtblks,                                       ! DMT SIZE IN PAGES
;  341        0837 1              dmtvbn,                                        ! VBN OF DMT IMAGE SECTION
;  342        0838 1              shrimgrab               : $rab (rac=seq),      ! RAB FOR READING SECTIONS OF SHAREABLE IMAGES
;  343        0839 1              curvbn,                                        ! CURRENT VBN FOR IMAGE SECTION
;  344        0840 1              shrimgclu               : ref block [,byte],   ! LAST OPEN SHAREABLE IMAGE TO CLUSTER DESC PTR
;  345        0841 1              dstvirtop,                                     ! AND THE TOP
;  346        0842 1              dstbuffer               : vector [2],          ! BUFFER DESCRIPTOR FOR DST GENERATED
;  347        0843 1              header_desc             : vector [2],          ! DESCRIPTOR OF HEADER BUFFER
;  348        0844 1              header_ptr              : initial (0),         ! RUNNING HEADER OUPUT POINTER
;  349        0845 1              header_size             : initial              ! ACCUMULATOR FOR IMAGE HEADER - STARTS
;  350        0846 1                                        (ihd$k_maxlength) ;  ! INCLUDING FIXED PART, ACTIVATION PART
;  351        0847 1      literal
;  352        0848 1              first_block_size        =  508,                ! USEABLE BYTES IN FIRST BLOCK
;  353        0849 1                                                             ! (The last word has a special meaning and the secon
;  354        0850 1                                                             !  and the second to the last word can't be used due
;  355        0851 1                                                             !  to a bug in the image-activator.)
;  356        0852 1              subseq_block_size       =  508,                ! USEABLE BYTES IN ALL BUT FIRST BLOCK OF HEADER
;  357        0853 1                                                             ! (The last word can't be used due to a bug in the
;  358        0854 1                                                             ! image-activator.)
;  359        0855 1              tfrlength               = 5120 ;               ! MAXIMUM TRANSFER BYTE COUNT
```

```
       1 global routine lnk$imginit : novalue =
       2 begin
       2 !
       2 !       THIS ROUTINE IS CALLED TO INITIALIZE THE IMAGE GENERATION
       2 !
       2 bind
       2     auxfnb = lnk$gl_imgfil [fdb$t_auxfnb] : block [nam$c_bln, byte] ; ! NAME THE NAM BLOCK
       2 local
       2     status,
       2     imgfab : block [fab$c_bln, byte] ;
       2
     P 2 $fab_init          (fab = imgfab                                            ! INITIALIZE THE FAB
     P 2                    ,fns = .lnk$gl_imgfil [fdb$w_usrnamlen]
     P 2                    ,fna = .lnk$gl_imgfil [fdb$l_usrnamadr]
     P 2                    ,dns = (if .lnk$gb_locnov_exe
     P 2                             then %charcount ('.EXE')
     P 2                             else %charcount ('SYS$DISK:[].EXE')
     P 2                            )
     P 2                    ,dna = (if .lnk$gb_locnov_exe
     P 2                             then uplit (byte ('.EXE'))
     P 2                             else uplit (byte ('SYS$DISK:[].EXE'))
     P 2                            )
     P 2                    ,nam = auxfnb
       2                    ) ;
       2
       2 if  .lnk$gb_locnov_exe                                                       ! USE 'OUTPUT FILE PARSING' UNLESS
       2 then imgfab [fab$v_ofp] = false                                              ! /EXE WAS A LOCAL QUALIFIER
       2 else imgfab [fab$v_ofp] = true ;                                             ! WITHOUT A SPECIFED VALUE
       2
       2 auxfnb [nam$l_rlf] = .lnk$gl_relnam_exe ;                                    ! SET RELATED FILE NAME ADDRESS
       2
       3 if $parse (fab = imgfab)                                                     ! PARSE THE FILENAME
       3 then begin
       3      if  .lnk$gt_imgnam [0] eql 0                                            ! NEED AN IMAGE NAME?
       4      then begin
       4           lnk$gt_imgnam [0] = .auxfnb [nam$b_name] ;                         ! SET LENGTH OF FILENAME
       4           ch$move (.lnk$gt_imgnam [0]                                        ! AND MOVE IT IN
       4                   ,.auxfnb [nam$l_name]
       4                   , lnk$gt_imgnam [1]
       4                   ) ;
       3           end ;
       3      if  .lnk$gt_imgid [0] eql 0                                             ! NEED AN IDENT?
       4      then begin
       4           lnk$gt_imgid [0] = .auxfnb [nam$b_type] ;                          ! FIGURE THE LENGTH
       4           ch$move (.lnk$gt_imgid [0]                                         ! AND MOVE IT IN
       4                   ,.auxfnb [nam$l_type]
       4                   , lnk$gt_imgid [1]
       4                   ) ;
       3           end ;
       3      ch$move (dsc$c_s_bln, lnk$filnamdsc (imgfab), lnk$gl_imgfil [fdb$q_filename]) ;
       2      end ;
       2
       2 if not .lnk$gl_ctlmsk [lnk$v_sys]                                            ! IF NOT A SYSTEM IMAGE,
       2 then if .lnk$gl_ctlmsk [lnk$v_trace]                                         ! AND TRACEBACK
       2      or .lnk$gl_ctlmsk [lnk$v_dbg]                                           ! OR DEBUGGING ENABLED, THEN
       3      then begin
       3           lnk$gl_dstend  = 0 ;                                               ! INITIALIZE END POINT AT START
```

```
;  418        0913 3              dstvirtop       = (.lnk$gl_dbgestim or 511) + 1 ;          ! AND ADD THE SIZE INTO VIRTUAL DESCRIPTOR
;  419        0914 3              lnk$gl_dstlocn = 0 ;                                       ! SET LOCATION COUNTER AT START
;  420        0915 3              status = lnk$requestmem (.dstvirtop/512, dstbuffer) ;      ! ALLOCATE A BUFFER FOR IT
;  421        0916 3              if not .status                                             !
;  422        0917 4              then  begin                                                ! BUT IF IT FAILS
;  423        0918 4                    lnk$gl_ctlmsk [lnk$v_image] = false ;                ! DISABLE IMAGE PRODUCTION
;  424        0919 4                    signal (.status, 0, lin$_noimgfil) ;                 ! REPORT THE ERROR
;  425        0920 4                    return ;                                           ! AND GIVE UP
;  426        0921 3                    end ;
;  427        0922 2              end ;
;  428        0923 2
;  429        0924 3 if (.lnk$gl_ctlmsk [lnk$v_sys] and .lnk$gl_ctlmsk [lnk$v_sysheadr])
;  430        0925 2 then lnk$gw_misects = 0 ;                                               ! DISABLE IMAGE COMPRESSION
;  431        0926 2
;  432        0927 2 if  .lnk$gl_gsbufdsc [0] neq 0                                          ! IF ANY GLOBAL SECTIONS ARE TO BE COPIED
;  433        0928 3 then if not (status = lnk$requestmem (.lnk$gl_gsbufdsc [0]              ! THEN ALLOCATE A BUFFER EQUAL IN SIZE
;  434        0929 3                                           , lnk$gl_gsbufdsc [1]         ! TO THE LARGEST ONE SEEN
;  435        0930 3           )                               )
;  436        0931 3     then begin                                                          ! BUT IF IT FAILS
;  437        0932 3              lnk$gl_ctlmsk [lnk$v_image] = false ;                      ! DISABLE IMAGE PRODUCTION
;  438        0933 3              signal (.status, 0, lin$_noimgfil) ;                       ! REPORT THE ERROR
;  439        0934 3              return ;                                                   ! AND GIVE UP
;  440        0935 2              end ;
;  441        0936 2 return ;
;  442        0937 1 end ;


                                                                      .TITLE  LNK_IMGOUT
                                                                      .IDENT  \V04-000\

                                                                      .PSECT  $PLIT$,NOWRT,NOEXE,2

                                          45  58  45  2E  00000 P.AAA:  .ASCII  \.EXE\                              ;
45  58  45  2E  5D  5B  3A  4B  53  49  44  24  53  59  53  00004 P.AAB:  .ASCII  \SYS$DISK:[].EXE\                   ;

                                                                      .PSECT  $OWN$,NOEXE,2

                                                00000 DMTBLKS:.BLKB   4
                                                00004 DMTVBN: .BLKB   4
                                           01   00008 SHRIMGRAB:
                                                              .BYTE   1
                                           44   00009         .BYTE   68                                         ;
                                         0000   0000A         .WORD   0                                          ;
                                     00000000   0000C         .LONG   0                                          ;
                                     00000000   00010         .LONG   0                                          ;
                                     00000000   00014         .LONG   0                                          ;
                                        0000#   00018         .WORD   0[3]                                       ;
                                         0000   0001E         .WORD   0                                          ;
                                     00000000   00020         .LONG   0                                          ;
                                         0000   00024         .WORD   0                                          ;
                                           00   00026         .BYTE   0                                          ;
                                           00   00027         .BYTE   0                                          ;
                                         0000   00028         .WORD   0                                          ;
                                         0000   0002A         .WORD   0                                          ;
                                     00000000   0002C         .LONG   0                                          ;
                                     00000000   00030         .LONG   0                                          ;
                                     00000000   00034         .LONG   0                                          ;
```

```
00000000  00038             .LONG   0                                  ;
      00  0003C             .BYTE   0                                  ;
      00  0003D             .BYTE   0                                  ;
      00  0003E             .BYTE   0                                  ;
      00  0003F             .BYTE   0                                  ;
00000000  00040             .LONG   0                                  ;
00000000  00044             .LONG   0                                  ;
00000000  00048             .LONG   0                                  ;
          0004C CURVBN:     .BLKB   4
          00050 SHRIMGCLU:
                            .BLKB   4
          00054 DSTVIRTOP:
                            .BLKB   4
          00058 DSTBUFFER:
                            .BLKB   8
          00060 HEADER_DESC:
                            .BLKB   8
00000000  00068 HEADER_PTR:
                            .LONG   0                                  ;
000000A8  0006C HEADER_SIZE:
                            .LONG   168                                ;

                            .PSECT  $GLOBAL$,NOEXE,2

      01  00000 LNK$AL_IMGRAB::
                            .BYTE   1                                  ;
      44  00001             .BYTE   68                                 ;
    0000  00002             .WORD   0                                  ;
00000000  00004             .LONG   0                                  ;
00000000  00008             .LONG   0                                  ;
00000000  0000C             .LONG   0                                  ;
   0000#  00010             .WORD   0[3]                               ;
    0000  00016             .WORD   0                                  ;
00000000  00018             .LONG   0                                  ;
    0000  0001C             .WORD   0                                  ;
      00  0001E             .BYTE   0                                  ;
      00  0001F             .BYTE   0                                  ;
    0000  00020             .WORD   0                                  ;
    0000  00022             .WORD   0                                  ;
00000000  00024             .LONG   0                                  ;
00000000  00028             .LONG   0                                  ;
00000000  0002C             .LONG   0                                  ;
00000000  00030             .LONG   0                                  ;
      00  00034             .BYTE   0                                  ;
      00  00035             .BYTE   0                                  ;
      00  00036             .BYTE   0                                  ;
      00  00037             .BYTE   0                                  ;
00000000  00038             .LONG   0                                  ;
00000000  0003C             .LONG   0                                  ;
00000000  00040             .LONG   0                                  ;
          00044 LNK$GL_IMGHDR::
                            .BLKB   4
          00048 LNK$GL_CURISD::
                            .BLKB   4
          0004C LNK$GL_IMGTOP::
                            .BLKB   4
          00050 LNK$GL_LOCN::
                 .BLKB   4
     00054 LNK$GL_DSTLOCN::
                 .BLKB   4
     00058 LNK$GL_DSTEND::
                 .BLKB   4
     0005C LNK$GL_RPROTISD::
                 .BLKB   4
     00060 LNK$GL_SHRCLSTRS::
                 .BLKB   4
     00064 LNK$GW_NCLUSTERS::
                 .BLKB   2
     00066 LNK$GW_DSTVBN::
                 .BLKB   2
     00068 LNK$GW_HDRBLKS::
                 .BLKB   2
     0006A LNK$GW_IMGBLKS::
                 .BLKB   2
     0006C LNK$GW_IMGIFI::
                 .BLKB   2
     0006E LNK$GW_DSTBLKS::
                 .BLKB   2
     00070 LNK$GW_GSTVBN::
                 .BLKB   2
     00072       .BLKB   2
     00074 LNK$GL_MATCHID::
                 .BLKB   4
FF   00078 LNK$GB_MATCHCTL::
                 .BYTE   -1                                                      ;

           ISD$C_SIZE==            88
           HDR$K_FILLCHR==         255
           IHD$K_SHR==             2
           IHD$K_ACTIVOFF==        48
           IHD$K_SYMDBGOFF==       68
           IHD$K_IMGIDOFF==        88
           IHD$K_PATCHOFF==        168
           IHD$K_MAXLENGTH==       168
           HDR$K_MINFILL==         2
                     .EXTRN  LNK$CLOSEFILE, LNK$REQUESTMEM
                     .EXTRN  LIB$TRAVERSE_TREE
                     .EXTRN  LNK$FILNAMDSC, LNK$ALLOBLK
                     .EXTRN  LNK$DEALBLK, LNK$ALLOVIRMEM
                     .EXTRN  LNK$CREFIXISD, LNK$SORTISECTS
                     .EXTRN  LNK$SEARCH, LIN$_CLOSEIN
                     .EXTRN  LIN$_CLOSEOUT, LIN$_CONFIXMEM
                     .EXTRN  LIN$_DBGTFR, LIN$_GETPTIERR
                     .EXTRN  LIN$_MEMFUL, LIN$_NOIMGFIL
                     .EXTRN  LIN$_OPENIN, LIN$_OPENOUT
                     .EXTRN  LIN$_OUTSIMG, LIN$_OUTSIMGP
                     .EXTRN  LIN$_READERR, LIN$_TFRSYS
                     .EXTRN  LIN$_USRTFR, LIN$_WRITEERR
                     .EXTRN  LNK$K_MAX_FILENAME_LENGTH
                     .EXTRN  LNK$GL_UNSUPPORTED
                     .EXTRN  LNK$GL_FILESLEFT
                     .EXTRN  LNK$GT_JPILST, LNK$GL_DMTBUFFER
                     .EXTRN  LNK$GL_DMTBYTES
                     .EXTRN  LNK$GL_MINVA, LNK$GL_LASTGADR
```

```
                                                        .EXTRN  LNK$GL_SPCGFIX, LNK$GL_SPCGIMGS
                                                        .EXTRN  LNK$GL_ADREFS, LNK$GL_SHRADR
                                                        .EXTRN  LNK$GL_SHRIMGS, LNK$GL_SHRSYMS
                                                        .EXTRN  LNK$GL_SHLEXTRA
                                                        .EXTRN  LNK$GL_FIXISD, LNK$GL_LOWCLU
                                                        .EXTRN  LNK$AW_VERSION, LNK$GT_SYSVER
                                                        .EXTRN  LNK$GL_CUROMD, LNK$GL_CURFIL
                                                        .EXTRN  LNK$GL_CURPSC, LNK$GL_INRELNAM
                                                        .EXTRN  LNK$GL_RELNAM_EXE
                                                        .EXTRN  LNK$GB_LOCNOV_EXE
                                                        .EXTRN  LNK$GB_PASS, LNK$GL_CTLMSK
                                                        .EXTRN  LNK$GL_CLULST, LNK$GL_CURCLU
                                                        .EXTRN  LNK$GL_LSTCLSTR
                                                        .EXTRN  LNK$GL_DEFCLU, LNK$GL_IMGFIL
                                                        .EXTRN  LNK$GL_DBGESTIM
                                                        .EXTRN  LNK$GL_DBGTFPS, LNK$GL_DBGTFR
                                                        .EXTRN  LNK$GW_GSTRECS, LNK$GL_TFRPSC
                                                        .EXTRN  LNK$GL_TFRADR, LNK$GQ_ALOSTIM
                                                        .EXTRN  LNK$GQ_PRIVS, LNK$GT_IMGNAM
                                                        .EXTRN  LNK$GT_IMGID, LNK$GW_CHANS
                                                        .EXTRN  LNK$GW_DZROMIN, LNK$GW_IOSEG
                                                        .EXTRN  LNK$GW_MISECTS, LNK$GW_NISECTS
                                                        .EXTRN  LNK$GW_SHRISCTS
                                                        .EXTRN  LNK$GW_NSYMBOLS
                                                        .EXTRN  LNK$GW_LSYMBOLS
                                                        .EXTRN  LNK$GL_GSBUFDSC
                                                        .EXTRN  SYS$PARSE

                                                        .PSECT  $CODE$,NOWRT,2

                                   0FFC 00000           .ENTRY  LNK$IMGINIT, Save R2,R3,R4,R5,R6,R7,R8,R9,-  ; 0856
                                                                R10,R11                                      :
                       5B 00000000'  EF  9E 00002       MOVAB   DSTVIRTOP, R11                               :
                       5A 00000000G  00  9E 00009       MOVAB   LNK$GT_IMGID, R10                            :
                       59 00000000G  00  9E 00010       MOVAB   LNK$GT_IMGNAM, R9                            :
                       58 00000000G  00  9E 00017       MOVAB   LNK$GL_CTLMSK, R8                            :
                       5E        B0  AE  9E 0001E       MOVAB   -80(SP), SP                                  :
                       57 00000000G  00  D0 00022       MOVL    LNK$GL_IMGFIL, R7                            ; 0862
                       56        26  A7  9E 00029       MOVAB   38(R7), R6                                   :
 0050   8F          00 6E            00  2C 0002D       MOVC5   #0, (SP), #0, #80, $RMS_PTR                  ; 0879
                                     6E     00034                                                            :
                       6E      5003  8F  B0 00035       MOVW    #20483, $RMS_PTR                             :
                  16   AE            02  90 0003A       MOVB    #2, $RMS_PTR+22                              :
                  1F   AE            02  90 0003E       MOVB    #2, $RMS_PTR+31                              :
                  28   AE            56  D0 00042       MOVL    R6, $RMS_PTR+40                              :
                  2C   AE        10  A7  D0 00046       MOVL    16(R7), $RMS_PTR+44                          :
                       51 00000000G  00  9A 0004B       MOVZBL  LNK$GB_LOCNOV_EXE, R1                        :
                       09            51  E9 00052       BLBC    R1, 1$                                       :
                       50 00000000'  EF  9E 00055       MOVAB   P.AAA, R0                                    :
                                     07  11 0005C       BRB     2$                                           :
                       50 00000000'  EF  9E 0005E 1$:   MOVAB   P.AAB, R0                                    :
                  30   AE            50  D0 00065 2$:   MOVL    R0, $RMS_PTR+48                              :
                  34   AE        0C  A7  90 00069       MOVB    12(R7), $RMS_PTR+52                          :
                       05            51  E9 0006E       BLBC    R1, 3$                                       :
                       50            04  D0 00071       MOVL    #4, R0                                       :
                                     03  11 00074       BRB     4$                                           :
                       50            0F  D0 00076 3$:   MOVL    #15, R0                                      :
```

```
                      35  AE                 50  90 00079 4$:   MOVB    R0, $RMS_PTR+53                          :
                          06                 51  E9 0007D       BLBC    R1, 5$                                   : 0881
                      07  AE                 20  8A 00080       BICB2   #32, IMGFAB+7                            : 0882
                                             04  11 00084       BRB     6$                                       :
                      07  AE                 20  88 00086 5$:   BISB2   #32, IMGFAB+7                            : 0883
                      10  A6 00000000G       00  D0 0008A 6$:   MOVL    LNK$GL_RELNAM_EXE, 16(R6)                : 0885
                                             5E  DD 00092       PUSHL   SP                                       : 0887
           00000000G  00                     01  FB 00094       CALLS   #1, SYS$PARSE                            :
                      37                     50  E9 0009B       BLBC    R0, 9$                                   :
                                             69  95 0009E       TSTB    LNK$GT_IMGNAM                            : 0889
                                             0D  12 000A0       BNEQ    7$                                       :
                      69         3B          A6  90 000A2       MOVB    59(R6), LNK$GT_IMGNAM                    : 0891
                      50                     69  9A 000A6       MOVZBL  LNK$GT_IMGNAM, R0                        : 0892
  01    A9       4C   B6                     50  28 000A9       MOVC3   R0, @76(R6), LNK$GT_IMGNAM+1             : 0894
                                             6A  95 000AF 7$:   TSTB    LNK$GT_IMGID                             : 0897
                                             0D  12 000B1       BNEQ    8$                                       :
                      6A         3C          A6  90 000B3       MOVB    60(R6), LNK$GT_IMGID                     : 0899
                      50                     6A  9A 000B7       MOVZBL  LNK$GT_IMGID, R0                         : 0900
  01    AA       50   B6                     50  28 000BA       MOVC3   R0, @80(R6), LNK$GT_IMGID+1              : 0902
                                             5E  DD 000C0 8$:   PUSHL   SP                                       : 0905
           00000000G  00                     01  FB 000C2       CALLS   #1, LNK$FILNAMDSC                        :
                      51 00000000G           00  D0 000C9       MOVL    LNK$GL_IMGFIL, R1                        :
  14    A1            60                     08  28 000D0       MOVC3   #8, (R0), 20(R1)                         :
        3B            68                     03  E0 000D5 9$:   BBS     #3, LNK$GL_CTLMSK, 12$                   : 0908
        04       02   A8                     02  E0 000D9       BBS     #2, LNK$GL_CTLMSK+2, 10$                 : 0909
        2E            68                     06  E1 000DE       BBC     #6, LNK$GL_CTLMSK, 11$                   : 0910
        50 00000000G  00 000001FF            8F  C9 000E2 10$:  BISL3   #511, LNK$GL_DBGESTIM, R0                : 0913
                      6B         01          A0  9E 000EE       MOVAB   1(R0), DSTVIRTOP                         :
                         00000000'           EF  7C 000F2       CLRQ    LNK$GL_DSTLOCN                           : 0914
                                 04          AB  9F 000F8       PUSHAB  DSTBUFFER                                : 0915
        7E            6B 00000200            8F  C7 000FB       DIVL3   #512, DSTVIRTOP, -(SP)                   :
           00000000G  00                     02  FB 00103       CALLS   #2, LNK$REQUESTMEM                       :
                      52                     50  D0 0010A       MOVL    R0, STATUS                               :
                      2D                     52  E9 0010D       BLBC    STATUS, 14$                              : 0916
        0B            68                     03  E1 00110 11$:  BBC     #3, LNK$GL_CTLMSK, 13$                   : 0924
        06       03   A8                     01  E1 00114 12$:  BBC     #1, LNK$GL_CTLMSK+3, 13$                 :
                         00000000G           00  B4 00119       CLRW    LNK$GW_MISECTS                           : 0925
                      50 00000000G           00  D0 0011F 13$:  MOVL    LNK$GL_GSBUFDSC, R0                      : 0927
                                             29  13 00126       BEQL    15$                                      :
                         00000000G           00  9F 00128       PUSHAB  LNK$GL_GSBUFDSC+4                        : 0929
                                             50  DD 0012E       PUSHL   R0                                       : 0928
           00000000G  00                     02  FB 00130       CALLS   #2, LNK$REQUESTMEM                       :
                      52                     50  D0 00137       MOVL    R0, STATUS                               :
                      14                     52  E8 0013A       BLBS    STATUS, 15$                              :
                      68                     01  8A 0013D 14$:  BICB2   #1, LNK$GL_CTLMSK                        : 0932
                         00000000G           8F  DD 00140       PUSHL   #LIN$_NOIMGFIL                           : 0933
                                             7E  D4 00146       CLRL    -(SP)                                    :
                                             52  DD 00148       PUSHL   STATUS                                   :
           00000000G  00                     03  FB 0014A       CALLS   #3, LIB$SIGNAL                           :
                                                 04 00151 15$:  RET                                              : 0937

; Routine Size:  338 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 global routine lnk$wrtimgbyts (bytecount, buffradr, reptcount, pos, siz) =
1 !
1 !     This routine is called to write "BYTECOUNT" bytes from "BUFFRADR" with
1 !     "REPTCOUNT" repetitions to the output image, starting at the current
1 !     location, LNK$GL_LOCN.  The location counter is updated by
1 !     BYTECOUNT*REPTCOUNT .
1 !
1 !     If "POS" AND "SIZ" are non-null parameters, the call is assumed to be
1 !     a variable position and size bit store.
1 !
1 !     Any attempt to write outside the image causes an error.
1 !
2 begin
2 local
2         status,
2         storelolimit,                                     ! LOWEST LEGAL OUTPUT ADDRESS
2         storehilimit,                                     ! HIGHEST LEGAL OUTPUT ADDRESS,
2         storehiaddr,                                      ! HIGHEST ADDRESS WE WILL OUTPUT TO
2         bitstore,                                         ! TRUE IF STORING VARIABLE POSITION AND SIZE
2         partition,                                        ! POINTER TO IMAGE PARTITION NAME (DS OR EX
2         storeaddr ;                                       ! OUTPUT ADDRESS
2
2 builtin
2         ap ;
2
2 bind
2         arglist = .ap : vector ;
2
2
2 if      .bytecount eql 0                                  ! IF ZERO BYTE COUNT
2 or      .reptcount eql 0                                  ! OR ZERO REPEAT COUNT
2 or not .lnk$gl_ctlmsk [lnk$v_image]                       ! OR NO IMAGE TO BE PRODUCED
2 then return true ;                                        ! RETURN DONE
2
2 bitstore = .(arglist [0])<0,8> eql 5 ;                    ! DETERMINE IF VARIABLE BIT STORE
2
2 storehiaddr = .lnk$gl_locn + .bytecount*.reptcount ;      ! FIGURE HIGHEST ADDRESS WE WANT TO STORE IN
2 storelolimit = 0 ;                                        ! LIMITS ARE INITIALLY 0
2 storehilimit = 0 ;                                        !
2 partition    = cstring('image binary') ;
2
2 if .lnk$gl_ctlmsk [lnk$v_dbgrec]                          ! IF THIS IS DEBUGGER DATA
2 or .bitstore                                              ! OR ITS A VARIABLE BIT STORE COMMAND
2 or  not ch$fail(ch$find_not_ch(.bytecount,.buffradr,0))   ! OR THERE IS A NON ZERO CHARACTER
3 or (if .lnk$gl_curisd eql 0                               ! IF NO ISECT SET UP
3     then false
3     else if .lnk$gl_curisd [isl$l_bufadr] eql 0           ! OR IF NO BUFFER SET UP
4          then begin                                       ! THEN WE ARE DONE
4               lnk$gl_locn = .storehiaddr ;                ! SO SET THE LOCATION COUNTER
4               return true ;                               ! AND RETURN SUCCESS
4               end
3         else true                                         ! NO NON-ZERO BYTES TO STORE
3    )                                                      ! BUT OTHERWISE OK
3 then begin                                                ! IT HAS BEEN TOUCHED BEFORE
3         !
3         !     THEN WE ARE GOING TO WRITE IT TO THE IMAGE SECTION
3         !
```

```
;  501          0995  3       if not .lnk$gl_ctlmsk [lnk$v_dbgrec]                    ! IF NOT A DEBUG DATA BYTE STREAM
;  502          0996  4       then begin                                              ! SET UP THE LO AND HI ADDRESS
;  503          0997  4            if .lnk$gl_curisd neq 0                            ! IF A IMAGE SECTION HAS BEEN
;  504          0998  5            then  begin                                        ! SET UP
;  505          0999  5                  bind hdrisd  = lnk$gl_curisd [isl$t_hdrisd]  ! LABEL FOR PART GOING TO HEADER
;  506          1000  5                                              : block [,byte] ;       !
;  507          1001  5                  storelolimit = .hdrisd [isd$v_vpg] ^ 9 ;             ! LOW LIMIT IS BASE OF ISECT
;  508          1002  5                  storehilimit = .storelolimit  +                      ! AND HIGH LIMIT IS BYTE BEYOND
;  509          1003  5                                .hdrisd [isd$w_pagcnt]*512 ;
;  510          1004  5                  if  .storehiaddr gtru .storehilimit                           ! IF WE WILL EXCEED IMAGE SE
;  511          1005  5                  and .lnk$gl_curpsc [psc$l_base]+.lnk$gl_curpsc [psc$l_length]  ! AND PSECT EXCEEDS IMAGE SE
;  512          1006  5                       gtru .storehilimit
;  513          1007  6                  then begin                                                    ! THEN WE MUST STORE AS SECT
;  514          1008  6                       local       bytes2write,                                 ! BYTES LEFT TO WRITE
;  515          1009  6                                   curisd ;                                     ! SAVED CURRENT ISD POINTER
;  516          1010  6                       curisd = .lnk$gl_curisd ;                                ! SAVE CURRENT ISECT DESCR.
;  517          1011  6                       if .lnk$gl_locn gtru .storelolimit                       ! IF ALREADY INTO ANOTHER IS
;  518          1012  7                       then begin                                               ! THEN FIND STARTING ISECT
;  519          1013  8                            while (lnk$gl_curisd = .lnk$gl_curisd [isl$l_nxtisd]) ! WHILE THERE ARE MORE ISECT
;  520          1014  7                                  neq 0
;  521          1015  8                            do begin
;  522          1016  8                               bind hdrisd  = lnk$gl_curisd [isl$t_hdrisd] : block [,byte] ; ! NAME PART FOR H
;  523          1017  8                               storelolimit = .hdrisd [isd$v_vpg] ^ 9 ;                       ! COMPUTE LOW LIMI
;  524          1018  8                               storehilimit = .storelolimit + .hdrisd [isd$w_pagcnt]*512 ;  ! AND HI LIMIT
;  525          1019  8                               if .storehiaddr lequ .storehilimit then exitloop ;
;  526          1020  7                               end ;
;  527          1021  7
;  528          1022  7                            if .lnk$gl_curisd eql 0 then return false ;              ! ERROR IF NO ISECT FOUND
;  529          1023  6                            end ;                                                    ! FINDING STARTING ISECT
;  530          1024  6
;  531          1025  6                       bytes2write = .bytecount * .reptcount ;                  ! FIGURE BYTES TO WRITE
;  532          1026  6                       while .bytes2write neq 0                                 ! WHILE THERE ARE MORE TO DO
;  533          1027  7                       do begin
;  534          1028  7                          local    roomleft,                                    ! ROOM LEFT IN THE CURRENT I
;  535          1029  7                                   isdrept,                                     ! REPEAT COUNT THIS ISD
;  536          1030  7                                   isdleft ;                                    ! BYTES LEFT FROM NON-FULL R
;  537          1031  7
;  538          1032  7                          roomleft = .storehilimit - .lnk$gl_locn ;             ! FIGURE ROOM LEFT IN IMAGE
;  539          1033  7                          while .roomleft neq 0                                 ! WHILE THERE IS ROOM IN THI
;  540          1034  7                                  and
;  541          1035  7                                 .bytes2write neq 0                             ! AND THERE ARE MORE BYTES T
;  542          1036  8                          do begin
;  543          1037  8                             isdrept     = minu (.roomleft/.bytecount,.reptcount) ; ! FIGURE NUMBER FULL REPEA
;  544          1038  8                             bytes2write = .bytes2write - .bytecount*.isdrept ;   ! ADJUST BYTE COUNT
;  545          1039  8                             isdleft     = .roomleft - .bytecount*.isdrept ;      ! AND NUMBER OF BYTES LEFT A
;  546          1040  8
;  547          1041  8                             if .bytes2write eql 0 then isdleft = 0 ;             ! NO PARTIAL BLOCK IF FINISH
;  548          1042  8
;  549          1043  8                             if not lnk$wrtimgbyts (.bytecount,.buffradr,.isdrept) ! WRITE THE FULL REPEATS
;  550          1044  8                             then return false ;
;  551          1045  8
;  552          1046  8                             if .isdleft neq 0                                    ! IF THERE IS A PARTIAL WRIT
;  553          1047  8                             then if not lnk$wrtimgbyts (.isdleft,.buffradr,1)    ! THEN WRITE IT OUT NOW
;  554          1048  8                                  then return false
;  555          1049  8                                  else bytes2write = .bytes2write - .isdleft ;    ! ADJUST COUNT
;  556          1050  8
;  557          1051  8                             roomleft = .roomleft - .bytecount*.isdrept - .isdleft ; ! ADJUST THE ROOM LEFT TH
;  558          1052   7                            end ;                                                  ! WHILE ROOM AND BYTES TO WR
;  559          1053   7
;  560          1054   7                         lnk$gl_curisd = .lnk$gl_curisd [isl$l_nxtisd] ;           ! LINK TO NEXT ISD
;  561          1055   7
;  562          1056   8                             begin
;  563          1057   8                             bind hdrisd = lnk$gl_curisd [isl$t_hdrisd]            ! NAME PART GOING TO HEADER
;  564          1058   8                                            : block [,byte] ;
;  565          1059   8                             if .isdleft neq 0                                     ! IF WE WROTE A PARTIAL STRI
;  566          1060   8                             then if not lnk$wrtimgbyts   (.bytecount-.isdleft     ! THEN WRITE THE REST OF THE
;  567          1061   8                                                          ,.buffradr +.isdleft
;  568          1062   8                                                          ,1
;  569          1063   8                                                          )
;  570          1064   8                                  then return false ;
;  571          1065   8
;  572          1066   8                             storelolimit = .hdrisd [isd$v_vpg] ^ 9 ;              ! SET NEW LOW LIMIT
;  573          1067   8                             storehilimit = .storelolimit +                        ! AND NEW HIGH LIMIT
;  574          1068   8                                                  .hdrisd [isd$w_pagcnt]*512 ;     !
;  575          1069   7                             end ;
;  576          1070   6                         end ;                                                     ! BYTES2WRITE NEQ 0
;  577          1071   6
;  578          1072   6                      lnk$gl_curisd = .curisd ;                                    ! RESTORE CURISD
;  579          1073   6                      return true ;                                                ! ALL DONE, RETURN SUCCESS
;  580          1074   5                      end ;                                                        ! EXCEEDING IMAGE SECTION
;  581          1075   4                 end ;                                                             ! POSITION TO WRITE BYTE STR
;  582          1076   4
;  583          1077   4         if .lnk$gl_curisd neq 0
;  584          1078   5         then begin
;  585          1079   5              bind hdrisd = lnk$gl_curisd [isl$t_hdrisd] : block [,byte] ;
;  586          1080   5
;  587          1081   5              if  .lnk$gl_curisd [isl$l_bufadr] eql 0                      ! IF A BUFFER HAS NOT YET BEEN
;  588          1082   5              and .hdrisd [isd$w_pagcnt] gtru 0                            ! AND ISECT IS NON-NULL
;  589          1083   6              then if not (status = lnk$requestmem (.hdrisd [isd$w_pagcnt] ! ALLOCATED, EXPAND MY PROGRAM REG
;  590          1084   6                                                    , lnk$gl_curisd [isl$l_bufadr]
;  591          1085   6                          )                         )
;  592          1086   6                   then  begin                                             ! BY SIZE OF I-SECT BEING WRITTEN
;  593          1087   6                         signal (.status, 0, lin$_noimgfil) ;              ! REPORT THE ERROR
;  594          1088   6                         lnk$gl_ctlmsk [lnk$v_image] =  false ;            ! SPACE AND TURN OFF ANY MORE
;  595          1089   6                         return true ;                                     ! ATTEMPTS
;  596          1090   5                         end ;
;  597          1091   5              storeaddr = .lnk$gl_curisd [isl$l_bufadr] ;                  ! START STORE ADDRESS COMPUTATION
;  598          1092   5              end
;  599          1093   4      end
;  600          1094   4 else begin                                                       ! BUT IF IT IS THE DST BEING WRITTEN
;  601          1095   4      storelolimit = 0 ;                                          ! THEN LIMITS ARE THE BASE AND TOP OF
;  602          1096   4      storehilimit = .dstvirtop ;                                 ! THE DST AS IF IT WERE MAPPED INTO THE EXE
;  603          1097   4      partition    = cstring('Debugger Symbol Table') ;           ! AT THE END
;  604          1098   4      storeaddr    = .dstbuffer [0] ;                             ! START COMPUTATION OF ACTUAL STORE ADDRESS
;  605          1099   3      end ;
;  606          1100   2 end ;
;  607          1101   2
;  608          1102   2 if .lnk$gl_locn lssu .storelolimit                               ! IF LOCATION COUNTER
;  609          1103   2 or .storehiaddr gtru .storehilimit                               ! BEYOND IMAGE
;  610          1104   3 then begin                                                       ! ALLOCATION, ISSUE AN
;  611          1105   3      if .lnk$gl_curpsc eql 0                                     ! IF PSECT NOT AVAILABLE
;  612          1106   3      then signal (lin$_outsimg,6,.lnk$gl_locn,                   ! ERROR AND GIVE UP
;  613          1107   3                           .partition,.storelolimit,.storehilimit,
;  614          1108   3                           lnk$gl_curomd [omd$b_namlng],
;  615          1109  3                         lnk$gl_curfil [fdb$q_filename],lin$_noimgfil)
;  616          1110  3         else signal (lin$_outsimgp,7,.lnk$gl_locn,                      ! ERROR AND GIVE UP
;  617          1111  3                         .partition,.storelolimit,.storehilimit,
;  618          1112  3                         lnk$gl_curpsc [psc$b_namlng],
;  619          1113  3                         lnk$gl_curomd [omd$b_namlng],
;  620          1114  3                         lnk$gl_curfil [fdb$q_filename],lin$_noimgfil) ;
;  621          1115  3         lnk$gl_ctlmsk [lnk$v_image] = false ;                           ! TURNING OFF ANY
;  622          1116  3         return false ;                                                  ! MORE ATTEMPTS
;  623          1117  2         end ;
;  624          1118  2
;  625          1119  2 if    .bitstore                                                        ! IF VARIABLE BIT STORE
;  626          1120  3 then begin
;  627          1121  3         bind imagebuf = .lnk$gl_curisd  [isl$l_bufadr] : block [,byte] ;
;  628          1122  3         map buffradr  : ref block [,byte] ;
;  629          1123  3
;  630          1124  3         imagebuf [(.lnk$gl_locn-.storelolimit), .pos, .siz, 0] = .buffradr ; ! DO THE BIT STORE
;  631          1125  3         return true ;
;  632          1126  2         end ;
;  633          1127  2
;  634          1128  2 storeaddr = .storeaddr + .lnk$gl_locn - .storelolimit ;               ! COMPUTE MAPPED
;  635          1129  2
;  636          1130  2 incr i from 1 to .reptcount                                            ! LOOP, MOVING THE REQUIRED BYTES,
;  637          1131  2 do    storeaddr = ch$move(.bytecount,.buffradr,.storeaddr) ;           ! UPDATING THE ADDRESS
;  638          1132  2
;  639          1133  2 lnk$gl_locn = .storehiaddr ;                                           ! UPDATE LOCATION COUNTER
;  640          1134  2
;  641          1135  2 if    .lnk$gl_ctlmsk [lnk$v_dbgrec]                                           ! IF THIS IS A DEBUG RECORD
;  642          1136  2 and (.lnk$gl_locn-1) gtru .lnk$gl_dstend                              ! AND THIS IS NEW HIGH DST ADDRESS
;  643          1137  2 then  lnk$gl_dstend = .lnk$gl_locn ;
;  644          1138  2
;  645          1139  2 return true ;                                                          ! AND ALL DONE
;  646          1140  1 end ;


                                                                      .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                        0C  00013 P.AAC:  .BYTE   12                                       ;
         79  72  61  6E  69  62  20  65  67  61  6D     69  00014         .ASCII  \image binary\                           ;
                                                        15  00020 P.AAD:  .BYTE   21                                       ;
6C  6F  62  6D  79  53  20  72  65  67  67  75  62  65  44  00021         .ASCII  \Debugger Symbol Table\                  ;
                                        65  6C  62  61  54  20  00030                                                      ;


                                                                      .PSECT  $CODE$,NOWRT,2

                                               0FFC 00000         .ENTRY  LNK$WRTIMGBYTS, Save R2,R3,R4,R5,R6,R7,R8,-  ; 0938
                                                                          R9,R10,R11                                   ;
                               5E              08  C2 00002         SUBL2   #8, SP                                       ;
                               59        04    AC  D0 00005         MOVL    BYTECOUNT, R9                                ; 0967
                                               66  13 00009         BEQL    4$                                           ;
                                         0C    AC  D5 0000B         TSTL    REPTCOUNT                                    ; 0968
                                               61  13 0000E         BEQL    4$                                           ;
                               5A 00000000G    00  E9 00010         BLBC    LNK$GL_CTLMSK, 4$                            ; 0969
                                               50  D4 00017         CLRL    R0                                           ; 0972
                               05              6C  91 00019         CMPB    (AP), #5                                     ;
```

```
                                            02  12 0001C          BNEQ    1$                                            ;
                                            50  D6 0001E          INCL    R0                                            ;
                        04    AE            50  D0 00020 1$:      MOVL    R0, BITSTORE                                  ;
               53             59       0C   AC  C5 00024          MULL3   REPTCOUNT, R9, R3                             ; 0974
                              6E 00000000'FF43  9E 00029          MOVAB   @LNK$GL_LOCN[R3], STOREHIADDR                 ;
                                            57  D4 00031          CLRL    STORELO[IMIT                                  ; 0975
                                            54  D4 00033          CLRL    STOREHILIMIT                                  ; 0976
                              5A 00000000'  EF  9E 00035          MOVAB   P.AAC, PARTITION                              ; 0977
52 00000000G   00             01            00  EF 0003C          EXTZV   #0, #1, LNK$GL_CTLMSK+2, R2                   ; 0979
                              2C            52  E8 00045          BLBS    R2, 5$                                        ;
                              28       04   AE  E8 00048          BLBS    BITSTORE, 5$                                  ; 0980
          08   BC             59            00  3B 0004C          SKPC    #0, R9, @BUFFRADR                             ; 0981
                                            02  12 00051          BNEQ    2$                                            ;
                                            51  D4 00053          CLRL    R1                                            ;
                                            51  D5 00055 2$:      TSTL    R1                                            ;
                                            1B  12 00057          BNEQ    5$                                            ;
                              50 00000000'  EF  D0 00059          MOVL    LNK$GL_CURISD, R0                             ; 0982
                                            03  12 00060          BNEQ    3$                                            ;
                                          01BC  31 00062          BRW     26$                                           ;
                                       08   A0  D5 00065 3$:      TSTL    8(R0)                                         ; 0984
                                            0A  12 00068          BNEQ    5$                                            ;
                  00000000'   EF            6E  D0 0006A          MOVL    STOREHIADDR, LNK$GL_LOCN                      ; 0986
                                          028B  31 00071 4$:      BRW     34$                                           ; 0987
                              03            52  E9 00074 5$:      BLBC    R2, 6$                                        ; 0995
                                          0190  31 00077          BRW     25$                                           ;
                              51 00000000'  EF  D0 0007A 6$:      MOVL    LNK$GL_CURISD, R1                             ; 0997
                                            2F  13 00081          BEQL    7$                                            ;
                              50       18   A1  9E 00083          MOVAB   24(R1), R0                                    ; 0999
57        04   A0             17            00  EF 00087          EXTZV   #0, #23, 4(R0), STORELOLIMIT                  ; 1001
               57             57            09  78 0008D          ASHL    #9, STORELOLIMIT, STORELOLIMIT                ;
                              50       02   A0  3C 00091          MOVZWL  2(R0), R0                                     ; 1003
               50             50            09  78 00095          ASHL    #9, R0, R0                                    ;
               54             50            57  C1 00099          ADDL3   STORELOLIMIT, R0, STOREHILIMIT                ;
                              54            6E  D1 0009D          CMPL    STOREHIADDR, STOREHILIMIT                     ; 1004
                                            10  1B 000A0          BLEQU   7$                                            ;
                              50 00000000G  00  D0 000A2          MOVL    LNK$GL_CURPSC, R0                             ; 1005
               50       18    A0       1C   A0  C1 000A9          ADDL3   28(R0), 24(R0), R0                            ;
                              54            50  D1 000AF          CMPL    R0, STOREHILIMIT                              ; 1006
                                            03  1A 000B2 7$:      BGTRU   8$                                            ;
                                          0103  31 000B4          BRW     22$                                           ;
                              5B            51  D0 000B7 8$:      MOVL    R1, CURISD                                    ; 1010
                              57 00000000'  EF  D1 000BA          CMPL    LNK$GL_LOCN, STORELOLIMIT                     ; 1011
                                            41  1B 000C1          BLEQU   12$                                           ;
                              51 00000000'  EF  D0 000C3          MOVL    LNK$GL_CURISD, R1                             ; 1013
                  00000000'   EF            61  D0 000CA 9$:      MOVL    (R1), [NK$GL_CURISD                           ;
                                            26  13 000D1          BEQL    10$                                           ; 1014
                              51 00000000'  EF  D0 000D3          MOVL    LNK$GL_CURISD, R1                             ; 1016
                              50       18   A1  9E 000DA          MOVAB   24(R1), R0                                    ;
57        04   A0             17            00  EF 000DE          EXTZV   #0, #23, 4(R0), STORELOLIMIT                  ; 1017
               57             57            09  78 000E4          ASHL    #9, STORELOLIMIT, STORELOLIMIT                ;
                              50       02   A0  3C 000E8          MOVZWL  2(R0), R0                                     ; 1018
               50             50            09  78 000EC          ASHL    #9, R0, R0                                    ;
               54             50            57  C1 000F0          ADDL3   STORELOLIMIT, R0, STOREHILIMIT                ;
                              54            6E  D1 000F4          CMPL    STOREHIADDR, STOREHILIMIT                     ; 1019
                                            D1  1A 000F7          BGTRU   9$                                            ;
                                 00000000'  EF  D5 000F9 10$:     TSTL    LNK$GL_CURISD                                 ; 1022
                                            03  12 000FF          BNEQ    12$                                           ;
```

```
                                                    01FF  31 00101 11$:    BRW     35$                                     ;
                                       55             53  D0 00104 12$:    MOVL    R3, BYTES2WRITE                         ; 1025
                                                      55  D5 00107 13$:    TSTL    BYTES2WRITE                             ; 1026
                                                      03  12 00109         BNEQ    14$                                     ;
                                                    00A3  31 0010B         BRW     21$                                     ;
                         52            54 00000000'   EF  C3 0010E 14$:    SUBL3   LNK$GL_LOCN, STOREHILIMIT, ROOMLEFT     ; 1032
                                                      54  13 00116 15$:    BEQL    19$                                     ; 1033
                                                      55  D5 00118         TSTL    BYTES2WRITE                             ; 1035
                                                      50  13 0011A         BEQL    19$                                     ;
                         50            52             59  C7 0011C         DIVL3   R9, ROOMLEFT, R0                        ; 1037
                                   0C  AC             50  D1 00120         CMPL    R0, REPTCOUNT                           ;
                                                      04  1B 00124         BLEQU   16$                                     ;
                                       50        0C   AC  D0 00126         MOVL    REPTCOUNT, R0                           ;
                                       58             50  D0 0012A 16$:    MOVL    R0, ISDREPT                             ;
                         50            59             58  C5 0012D         MULL3   ISDREPT, R9, R0                         ; 1038
                                       55             50  C2 00131         SUBL2   R0, BYTES2WRITE                         ;
                         56            52             50  C3 00134         SUBL3   R0, ROOMLEFT, R6                        ; 1039
                                       53             56  D0 00138         MOVL    R6, ISDLEFT                             ;
                                                      55  D5 0013B         TSTL    BYTES2WRITE                             ; 1041
                                                      02  12 0013D         BNEQ    17$                                     ;
                                                      53  D4 0013F         CLRL    ISDLEFT                                 ;
                                                      58  DD 00141 17$:    PUSHL   ISDREPT                                 ; 1043
                                                 08   AC  DD 00143         PUSHL   BUFFRADR                                ;
                                                      59  DD 00146         PUSHL   R9                                      ;
                                 FEB3  CF             03  FB 00148         CALLS   #3, LNK$WRTIMGBYTS                      ;
                                       B1             50  E9 0014D         BLBC    R0, 11$                                 ;
                                                      53  D5 00150         TSTL    ISDLEFT                                 ; 1046
                                                      12  13 00152         BEQL    18$                                     ;
                                                      01  DD 00154         PUSHL   #1                                      ; 1047
                                                 08   AC  DD 00156         PUSHL   BUFFRADR                                ;
                                                      53  DD 00159         PUSHL   ISDLEFT                                 ;
                                 FEA0  CF             03  FB 0015B         CALLS   #3, LNK$WRTIMGBYTS                      ;
                                       9E             50  E9 00160         BLBC    R0, 11$                                 ;
                                       55             53  C2 00163         SUBL2   ISDLEFT, BYTES2WRITE                    ; 1049
                         52            56             53  C3 00166 18$:    SUBL3   ISDLEFT, R6, ROOMLEFT                   ; 1051
                                                      AA  11 0016A         BRB     15$                                     ; 1033
                            00000000'  EF 00000000'   FF  D0 0016C 19$:    MOVL    @LNK$GL_CURISD, LNK$GL_CURISD           ; 1054
                         52 00000000'  EF             18  C1 00177         ADDL3   #24, LNK$GL_CURISD, R2                  ; 1057
                                                      53  D5 0017F         TSTL    ISDLEFT                                 ; 1059
                                                      15  13 00181         BEQL    20$                                     ;
                                                      01  DD 00183         PUSHL   #1                                      ; 1060
                                                 08 BC43  9F 00185         PUSHAB  @BUFFRADR[ISDLEFT]                      ; 1061
                         7E            59             53  C3 00189         SUBL3   ISDLEFT, R9, -(SP)                      ; 1060
                                 FE6E  CF             03  FB 0018D         CALLS   #3, LNK$WRTIMGBYTS                      ;
                                       03             50  E8 00192         BLBS    R0, 20$                                 ;
                                                    016B  31 00195         BRW     35$                                     ;
            57      04   A2            17             00  EF 00198 20$:    EXTZV   #0, #23, 4(R2), STORELOLIMIT            ; 1066
                         57            57             09  78 0019E         ASHL    #9, STORELOLIMIT, STORELOLIMIT          ;
                                       52        02   A2  3C 001A2         MOVZWL  2(R2), R2                               ; 1068
                         52            52             09  78 001A6         ASHL    #9, R2, R2                              ;
                         54            52             57  C1 001AA         ADDL3   STORELOLIMIT, R2, STOREHILIMIT          ;
                                                    FF56  31 001AE         BRW     13$                                     ; 1026
                            00000000'  EF             5B  D0 001B1 21$:    MOVL    CURISD, LNK$GL_CURISD                   ; 1072
                                                      40  11 001B8         BRB     23$                                     ; 1073
                                       50 00000000'   EF  D0 001BA 22$:    MOVL    LNK$GL_CURISD, R0                       ; 1077
                                                      5E  13 001C1         BEQL    26$                                     ;
                                       51        18   A0  9E 001C3         MOVAB   24(R0), R1                              ; 1079
```

```
                                      08  A0  D5  001C7       TSTL    8(R0)                                   ; 1081
                                          31  12  001CA       BNEQ    24$                                     ;
                                      02  A1  B5  001CC       TSTW    2(R1)                                   ; 1082
                                          2C  13  001CF       BEQL    24$                                     ;
                                      08  A0  9F  001D1       PUSHAB  8(R0)                                   ; 1084
                            7E        02  A1  3C  001D4       MOVZWL  2(R1), -(SP)                            ;
                 00000000G  00            02  FB  001D8       CALLS   #2, LNK$REQUESTMEM                      ;
                            1B            50  E8  001DF       BLBS    STATUS, 24$                             ;
                               00000000G  8F  DD  001E2       PUSHL   #LIN$_NOIMGFIL                          ; 1087
                                          7E  D4  001E8       CLRL    -(SP)                                   ;
                                          50  DD  001EA       PUSHL   STATUS                                  ;
                 00000000G  00            03  FB  001EC       CALLS   #3, LIB$SIGNAL                          ;
                 00000000G  00            01  8A  001F3       BICB2   #1, LNK$GL_CTLMSK                       ; 1088
                                        0102  31  001FA 23$:  BRW     34$                                     ; 1089
                            50 00000000'  EF  D0  001FD 24$:  MOVL    LNK$GL_CURISD, R0                       ; 1091
                            58        08  A0  D0  00204       MOVL    8(R0), STOREADDR                        ;
                                          17  11  00208       BRB     26$                                     ; 1077
                                          57  D4  0020A 25$:  CLRL    STORELOLIMIT                            ; 1095
                            54 00000000'  EF  D0  0020C       MOVL    DSTVIRTOP, STOREHILIMIT                 ; 1096
                            5A 00000000'  EF  9E  00213       MOVAB   P.AAD, PARTITION                        ; 1097
                            58 00000000'  EF  D0  0021A       MOVL    DSTBUFFER, STOREADDR                    ; 1098
                            52 00000000'  EF  D0  00221 26$:  MOVL    LNK$GL_LOCN, R2                         ; 1102
                            57            52  D1  00228       CMPL    R2, STORELOLIMIT                        ;
                                          05  1F  0022B       BLSSU   27$                                     ;
                            54            6E  D1  0022D       CMPL    STOREHIADDR, STOREHILIMIT               ; 1103
                                          65  1B  00230       BLEQU   30$                                     ;
              55 00000000G  00            14  C1  00232 27$:  ADDL3   #20, LNK$GL_CURFIL, R5                  ; 1109
              51 00000000G  00            28  C1  0023A       ADDL3   #40, LNK$GL_CUROMD, R1                  ; 1108
                            50 00000000G  00  D0  00242       MOVL    LNK$GL_CURPSC, R0                       ; 1105
                                          21  12  00249       BNEQ    28$                                     ;
                               00000000G  8F  DD  0024B       PUSHL   #LIN$_NOIMGFIL                          ; 1109
                                          22  BB  00251       PUSHR   #^M<RT,R5>                              ;
                                          54  DD  00253       PUSHL   STOREHILIMIT                            ;
                                          57  DD  00255       PUSHL   STORELOLIMIT                            ;
                                    0404  8F  BB  00257       PUSHR   #^M<R2,R10>                             ;
                                          06  DD  0025B       PUSHL   #6                                      ;
                               00000000G  8F  DD  0025D       PUSHL   #LIN$_OUTSIMG                           ;
                 00000000G  00            09  FB  00263       CALLS   #9, LIB$SIGNAL                          ;
                                          22  11  0026A       BRB     29$                                     ;
                               00000000G  8F  DD  0026C 28$:  PUSHL   #LIN$_NOIMGFIL                          ; 1114
                                          22  BB  00272       PUSHR   #^M<RT,R5>                              ;
                                      2D  A0  9F  00274       PUSHAB  45(R0)                                  ; 1112
                                          54  DD  00277       PUSHL   STOREHILIMIT                            ; 1114
                                          57  DD  00279       PUSHL   STORELOLIMIT                            ;
                                    0404  8F  BB  0027B       PUSHR   #^M<R2,R10>                             ;
                                          07  DD  0027F       PUSHL   #7                                      ;
                               00000000G  8F  DD  00281       PUSHL   #LIN$_OUTSIMGP                          ;
                 00000000G  00            0A  FB  00287       CALLS   #10, LIB$SIGNAL                         ;
                 00000000G  00            01  8A  0028E 29$:  BICB2   #1, LNK$GL_CTLMSK                       ; 1115
                                          6C  11  00295       BRB     35$                                     ; 1116
                            1D        04  AE  E9  00297 30$:  BLBC    BITSTORE, 31$                           ; 1119
                            50 00000000'  EF  D0  0029B       MOVL    LNK$GL_CURISD, R0                       ; 1121
              51 00000000'  EF            57  C3  002A2       SUBL3   STORELOLIMIT, LNK$GL_LOCN, R1           ; 1124
                            51        08  A0  C0  002AA       ADDL2   8(R0), R1                               ;
61       14   AC        10  AC        08  AC  F0  002AE       INSV    BUFFRADR, POS, SIZ, (R1)                ;
                                          47  11  002B6       BRB     34$                                     ; 1125
              53            58 00000000'  EF  C1  002B8 31$:  ADDL3   LNK$GL_LOCN, STOREADDR, R3              ; 1128
```

```
            58          53               57  C3 002C0        SUBL3   STORELOLIMIT, R3, STOREADDR         ;
                                         57  D4 002C4        CLRL    I                                   ; 1131
                                         08  11 002C6        BRB     33$                                 ;
            68        08  BC             59  28 002C8 32$:   MOVC3   R9, @BUFFRADR, (STOREADDR)          ;
                          58             53  D0 002CD        MOVL    R3, STOREADDR                       ;
            F3            57        0C   AC  F3 002D0 33$:   AOBLEQ  REPTCOUNT, I, 32$                   ;
              00000000'   EF             6E  D0 002D5        MOVL    STOREHIADDR, LNK$GL_LOCN            ; 1133
                          1C 00000000G   00  E9 002DC        BLBC    LNK$GL_CTLMSK+2, 34$                ; 1135
            50 00000000'  EF             01  C3 002E3        SUBL3   #1, LNK$GL_LOCN, R0                 ; 1136
               00000000'  EF             50  D1 002EB        CMPL    R0, LNK$GL_DSTEND                   ;
                                         0B  1B 002F2        BLEQU   34$                                 ;
               00000000'  EF 00000000'   EF  D0 002F4        MOVL    LNK$GL_LOCN, LNK$GL_DSTEND          ; 1137
                          50             01  D0 002FF 34$:   MOVL    #1, R0                              ; 1139
                                             04 00302        RET                                         ;
                                         50  D4 00303 35$:   CLRL    R0                                  ; 1140
                                             04 00305        RET                                         ;

; Routine Size:  774 bytes,    Routine Base:  $CODE$ + 0152


;  647          1141  1
```

```
1 global routine lnk$flushimg : novalue =
2 begin
2 !
2 !++
2 !       THIS ROUTINE IS CALLED WITHOUT ANY ARGUMENTS TO FLUSH THE
2 !       BINARY THAT WAS GENERATED TO AN IMAGE FILE.
2 !       THE FIRST PHASE IS TO GO DOWN THE LIST OF IMAGE SECTION DESCRIPTORS
2 !       AND WITH THE USE OF THE MEMORY MANAGEMENT SYSTEM SERVICE
2 !       $GETPTI, COMPRESSING THE UNINITIALIZED PAGES FROM EACH IMAGE SECTION
2 !       THE COMPRESSION IS ACHIEVED BY GENERATING A DEMAND ZERO IMAGE
2 !       SECTION DESCRIPTOR FOR EACH REGION OF AN IMAGE SECTION THAT
2 !       HAS NOT BEEN TOUCHED DURING PASS 2.
2 !       THE SIZES OF THE IMAGE HEADER AND IMAGE BINARY ARE ACCUMULATED DURING THIS
2 !       PHASE WHICH ENDS WITH CREATION OF THE FILE AND ALLOCATION OF A BUFFER FOR THE HEADER.
2 !       THE NEXT PHASE CONSISTS OF ANOTHER PASS DOWN THE LIST OF
2 !       IMAGE SECTION DESCRIPTORS, CREATING THE ISD ENTRY IN THE
2 !       HEADER (STILL IN MEMORY) AND WRITING TO THE IMAGE FILE THOSE
2 !       IMAGE SECTIONS THAT CONTAIN BINARY DATA.
2 !       FINALLY THE DEBUG SYMBOL TABLE, IF ANY IS WRITTEN TO THE FILE
2 !       AND THE SUMMARY PARAMETERS OF THE IMAGE HEADER ARE COMPUTED.
2 !       THE FILE IS LEFT OPEN FOR POSSIBLE APPENDING OF THE GLOBAL SYMBOL
2 !       TABLE FOLLOWED BY THE WRITING OF THE IMAGE HEADER.
2 !
2 !--
2
2 label
2       procsisd ;
2 !
2 local
2       based,                                          ! 1 IF IMAGE IS BASED, ZERO OTHERWISE
2       isdlist         : ref vector [,long],           ! POINTER TO TABLE IF ISD ADDRESSES
2       isdnum,                                         ! INDEX INTO ISECT LIST
2       shrisdlist      : ref vector [,long],           ! PTR TO TABLE OF ISD ADDRESSES IN PIC SHR IMAGES
2       shrisdnum,                                      ! INDEX INTO SHRISDLIST
2       shrclulst       : ref vector [,long],           ! LIST OF CLUSTER DESCRIPTORS
2       shrclunum,                                      ! INDEX IN SHRCLULST
2       adrclulst       : ref vector [,long],           ! LIST OF LOCAL CLUSTERS REFERENCED BY .ADDRESS
2       adrclunum,                                      ! INDEX INTO ADRCLULST
2       errorcode,                                      ! WRITE ERROR CODE RETURNED
2       rmsstv,                                         ! STV RETURNED FROM RMS
2       emptydesc       : vector [2],                   ! DESCRIPTOR RETURNED FOR EMPTY PAGES
2       fileblks,                                       ! NUMBER OF BLOCKS IN FILE
2       lastcluster,                                    ! TRUE IF CLUSTER IS LAST CLUSTER
2       imagefab        : block [fab$c_bln,byte],       ! FILE ACCESS BLOCK - JUST FOR CREATE
2       firstisd        : ref block [,byte],            ! POINTER TO FIRST ISD IN CLUSTER
2       previsd         : ref block [,byte] ;           ! POINTER TO PREVIOUS ISD
2
2 bind
2    auxfnb = lnk$gl_imgfil [fdb$t_auxfnb] : block [nam$c_bln,byte] ;    ! AUXILIARY FILENAME BLOCK
2
2 if      .lnk$gl_ctlmsk [lnk$v_sys]                                      ! IF A SYSTEM IMAGE
2 and not .lnk$gl_ctlmsk [lnk$v_sysheadr]                                 ! AND NO HEADER REQUESTED
3 then begin                                                              ! THERE IS ONLY ONE ISD (BUT NO HEAD
3     bind sysisd    = lnk$gl_defclu [clu$l_fstisd] : ref block [,byte],  ! GIVE IT A NAME
3          syshdrisd = sysisd [isl$t_hdrisd]        : block [,byte] ;     ! AND FOR PART THAT WOULD BE HEADER
3
3     header_size = 0 ;                                                   ! NO HEADER
```

```
;  706       1199 3      if .lnk$gl_defclu [clu$l_fstisd] neq 0                     ! IF THERE IS AN IMAGE SECTION
;  707       1200 3      then lnk$gw_imgblks = .syshdrisd [isd$w_pagcnt] ;          ! GET THE BLOCK COUNT
;  708       1201 3      end
;  709       1202 3 else begin                                                      ! ELSE, IF NO GSMATCH OPTION (IGNORED EXCEPT
;  710       1203 3      if .lnk$gb_matchctl  eql %x'FF'                            ! FOR SHAREABLE IMAGES)
;  711       1204 4      then begin                                                 ! THEN FORCE A MATCH EQUAL
;  712       1205 4           lnk$gb_matchctl = isd$k_matequ ;                      ! SINCE USER OPTED NOT TO SPECIFY
;  713       1206 4           lnk$gl_matchid  = .lnk$gq_alostim <16,31,0> ;         ! USE THE MIDDLE 31 BITS OF LINK TIME
;  714       1207 3           end ;                                                 ! (default high bit must be zero)
;  715       1208 3       lnk$gl_curclu = lnk$gl_clulst ;                           ! START AT FIRST CLUSTER
;  716       1209 3       while(lnk$gl_curclu = .lnk$gl_curclu [clu$l_nxtclu]) neq 0 ! GO DOWN THE LIST
;  717       1210 4       do begin
;  718       1211 5          lastcluster = (.lnk$gl_curclu eql .lnk$gl_lstclstr)    ! SEE IF THIS IS THE LAST CLUSTER
;  719       1212 4                      and not .lnk$gl_ctlmsk [lnk$v_shr] ;       !  BUT ONLY IN EXECUTABLE IMAGES
;  720       1213 4          previsd = lnk$gl_curclu [clu$l_fstisd] ;
;  721       1214 4          while (lnk$gl_curisd = .previsd [isl$l_nxtisd]) neq 0  ! WHILE THERE IS ANOTHER
;  722       1215 4          do
;  723       1216 5 procsisd:    begin
;  724       1217 5              bind curhdrisd = lnk$gl_curisd [isl$t_hdrisd]      ! NAME THE HEADER PART OF IT
;  725       1218 5                                        : block [,byte] ;
;  726       1219 5              if .lnk$gl_curisd eql .lnk$gl_fixisd               ! IF THIS IS THE FIXUP SECTION
;  727       1220 6              then begin
;  728       1221 6                   lnk$gw_imgblks = .lnk$gw_imgblks + .curhdrisd [isd$w_pagcnt] ; ! ADD IN COUNT
;  729       1222 6                   previsd        = .lnk$gl_curisd ;                              ! MOVE TO NEXT
;  730       1223 6                   leave procsisd ;                              ! AND DON'T TOUCH THE FIXUP SECTION
;  731       1224 5                   end ;
;  732       1225 5              if .curhdrisd [isd$w_pagcnt] eql 0                 ! IF ZERO LENGTH I-SECTION
;  733       1226 6              then begin
;  734       1227 6                   lnk$gw_nisects = .lnk$gw_nisects  - 1 ;       ! DISCOUNT
;  735       1228 6                   previsd        = .lnk$gl_curisd ;             ! MOVE TO NEXT
;  736       1229 6                   leave procsisd ;                              ! AND SKIP THIS ONE
;  737       1230 5                   end ;
;  738       1231 5
;  739       1232 5              if  .lnk$gl_ctlmsk [lnk$v_protect]                 ! IF IMAGE IS PROTECTED
;  740       1233 5              or  .lnk$gl_curclu [clu$v_protect]                 ! OR IMAGE IS PROTECTED
;  741       1234 5              or  .curhdrisd [isd$v_vector]                      ! OR ISECT IS A VECTOR ISECT
;  742       1235 5              then curhdrisd [isd$v_protect] = true ;            !   THEN PROTECT THE IMAGE SECTION
;  743       1236 5
;  744       1237 5              if .lastcluster
;  745       1238 5              then curhdrisd [isd$v_lastclu] = true ;
;  746       1239 5
;  747       1240 5              if  not .curhdrisd [isd$v_dzro]                    ! IF NOT ALREADY A DEMAND ZERO SECTION
;  748       1241 5              and .lnk$gl_curisd [isl$l_bufadr] eql 0            ! BUT NO BUFFER YET ALLOCATED
;  749       1242 5              then if      .curhdrisd [isd$v_wrt]
;  750       1243 5                   and not .curhdrisd [isd$v_gbl]                ! AND NOT GLOBAL
;  751       1244 6                   and not (.lnk$gl_ctlmsk [lnk$v_shr]           ! *** TEMP DUE TO IMAGE ACTIVATOR BUG ***
;  752       1245 6                                and not                          !   Use CRF instead of DZRO if we're buildin
;  753       1246 6                            .lnk$gl_unsupported [0]              !   a shareable image and the "unsupported"
;  754       1247 6                          )                                      !   DZRO option has not be selected
;  755       1248 6                   then begin
;  756       1249 6                        curhdrisd [isd$w_size] = isd$k_lendzro ; ! THEN CONVERT WHOLE SECTION
;  757       1250 6                        curhdrisd [isd$v_dzro] = true ;          ! TO A DEMAND ZERO
;  758       1251 6                        curhdrisd [isd$v_crf]  = false ;         ! DISABLING POSSIBLE COPY ON REFERENCE
;  759       1252 6                        end
;  760       1253 5                   else if  .curhdrisd [isd$v_gbl]               ! OTHERWISE, IF IT IS GLOBAL
;  761       1254 6                        then begin
;  762       1255 6                             if       .lnk$gl_curclu [clu$v_copy] ! AND THE CLUSTER IS TO BE
;  763          1256  6                   and not .curhdrisd [isd$v_protect]    ! COPIED INTO THIS IMAGE
;  764          1257  7                   then begin
;  765          1258  7                        lnk$gl_curisd [isl$l_bufadr] = .lnk$gl_gsbufdsc [1] ; ! SET THE BUFFER
;  766          1259  7                        lnk$gl_curisd [isl$l_bufend] = .lnk$gl_gsbufdsc [2] ; ! DESCRIPTOR
;  767          1260  7                        end
;  768          1261  7                   else begin
;  769          1262  7                        curhdrisd [isd$l_vbn]        = 0 ;               ! SET NO VBN
;  770          1263  7                        lnk$gl_curisd [isl$l_bufadr] = 0 ;               ! AND NO BUFFER
;  771          1264  7                        lnk$gl_curisd [isl$l_bufend] = 0 ;
;  772          1265  6                        end ;
;  773          1266  6                   end
;  774          1267  6              else begin
;  775          1268  6                   local status ;
;  776          1269  6                   status = lnk$requestmem (.curhdrisd [isd$w_pagcnt]    ! IF LOCAL SECTION, ALLOCAT
;  777          1270  6                                           ,lnk$gl_curisd [isl$l_bufadr] ! A BUFFER OF ZEROES
;  778          1271  6                                           ) ;
;  779          1272  6                    if not .status
;  780          1273  7                    then begin
;  781          1274  7                         signal (.status, 0, lin$_noimgfil) ;            ! REPORT THE ERROR
;  782          1275  7                         lnk$gl_ctlmsk [lnk$v_image] = false ;           ! DISABLE IMAGE PRODUCTION
;  783          1276  7                         return ;                                        ! AND GIVE UP
;  784          1277  6                         end ;
;  785          1278  5                    end ;
;  786          1279  5
;  787          1280  5     if .lnk$gl_curisd [isl$l_bufadr] neq 0                      ! IF THERE IS NOW A BUFFER
;  788          1281  5     and                                                         ! AND SECTION IS TOO SMALL
;  789          1282  6         (       .curhdrisd [isd$w_pagcnt] lssu .lnk$gw_dzromin ! FOR COMPRESSION OR
;  790          1283  6          or not .curhdrisd [isd$v_wrt]                          ! IS READ ONLY OR WE
;  791          1284  6          or     .lnk$gw_nisects gequ .lnk$gw_misects            ! HAVE EXCEEDED LIMIT ON ISD'S
;  792          1285  6         )                                                       ! THEN ADD THE SECTION
;  793          1286  5     then lnk$gw_imgblks = .lnk$gw_imgblks + .curhdrisd [isd$w_pagcnt] ; ! SIZE TO IMAGE
;  794          1287  5
;  795          1288  5    if     .curhdrisd [isd$v_dzro]                                  ! IF SECTION IS STILL DEMAND ZERO
;  796          1289  5    or not .curhdrisd [isd$v_wrt]                                   ! OR IS NOT WRITABLE
;  797          1290  5    or     .curhdrisd [isd$v_gbl]                                   ! OR THIS IS GLOBAL
;  798          1291  5    or     .curhdrisd [isd$w_pagcnt] lssu .lnk$gw_dzromin           ! OR THE SECTION IS TOO SMALL
;  799          1292  5    or     .lnk$gw_nisects gequ .lnk$gw_misects                     ! OR WE HAVE ENOUGH OF THEM
;  800          1293  6    or    (.lnk$gl_ctlmsk [lnk$v_shr]                               !** TEMP DUE TO IMAGE ACTIVATOR BUG
;  801          1294  6                and not                                             !  Don't convert to DZRO if building
;  802          1295  6          .lnk$gl_unsupported [0]                                   !  a shr image and the ''unsupported''
;  803          1296  6         )                                                          !  DZRO option has not be selected
;  804          1297  5    then previsd = .lnk$gl_curisd                                   ! WE ARE DONE WITH IT - THIS IS NOW
;  805          1298  6    else begin
;  806          1299  7         do  begin                                                 ! LOOP FOR CALLS TO PTI SERVICE
;  807          1300  7             bind curhdrisd = lnk$gl_curisd [isl$t_hdrisd]         ! RENAME THE HEADER PART
;  808          1301  7                                      : block [,byte] ;            ! FOR POSSIBLE CHANGE OF CURRENT ISD
;  809        P 1302  8             if not (errorcode = $getpti
;  810        P 1303  8                        (inadr  = lnk$gl_curisd [isl$l_bufadr]    ! CALL $GETPTI TO FIND UNTOUCHED PAG
;  811        P 1304  8                        ,retadr = emptydesc [0]                   ! TWO -LONGWORD RETURN ARRAY
;  812        P 1305  8                        ,acmode = 0                               ! ACCESS MODE IS USER
;  813        P 1306  8                        ,mask   = 1                               ! PAGES WHICH ARE DEMAND ZERO
;  814        P 1307  8                        ,pagcnt =.lnk$gw_dzromin                  ! MUST BE AT LEAST THIS MANY PAGES
;  815          1308  8                     )  )
;  816          1309  8             then begin                                            ! BUT IF AN ERROR RETURNED
;  817          1310  8                  signal  (lin$_getptierr,4                        ! THEN TELL THE USER AND
;  818          1311  8                         ,.lnk$gl_curisd [isl$l_bufadr]            ! CONTINUE WITHOUT THE IMAGE
;  819          1312  8                         ,.lnk$gl_curisd [isl$l_bufend]
8                   ..curhdrisd [isd$v_vpg]
8                   ..curhdrisd [isd$w_pagcnt]
8                   ,.errorcode, 0 ,lin$_noimgfil
8                   ) ;
8            lnk$gl_ctlmsk [lnk$v_image] = false ;              ! DISABLE ANY MORE IMAGE OUTPUT
8            return ;
7            end ;
7       if (emptydesc [1] = .emptydesc [1] + 1) eql 0           ! IF WE DON'T HAVE ANOTHER EMPTY
7       or .lnk$gw_nisects gequ .lnk$gw_misects                 ! REGION OR WE HAVE ENOUGH OF THEM
8       then begin
8            previsd        = .lnk$gl_curisd ;                  ! THEN CURRENT BECOMES PREVIOUS
8            lnk$gw_imgblks = .lnk$gw_imgblks + .curhdrisd [isd$w_pagcnt] ; ! AND TO THE IMAGE
8            emptydesc [1]  =  0 ;                              ! ENSURE END OF LOOP
8            end
8       else begin                                              ! IF WE DO HAVE ANOTHER ....
8            local  newisd    : ref block [,byte],              ! POINTER FOR NEWLY ALLOCATED ISD
8                   newhdrisd : ref block [,byte] ;             ! AND THE HEADER PART OF IT
8
8            lnk$alloblk (isd$c_size,newisd) ;                  ! ALLOCATE AN IMAGE SECTION DESCRIPT
8            newisd [isl$l_cludsc] = .lnk$gl_curclu ;           ! SET CLUSTER DESCRIPTOR POINTER
8            newisd [isl$l_bufadr] = 0 ;                        ! ZERO THE BUFFER ADDRESS
8            newisd [isl$l_bufend] = 0 ;                        ! QUADWORD
8            newhdrisd = newisd [isl$t_hdrisd] ;                ! SET POINTER TO HEADER PART
8            newhdrisd [isd$w_size] = isd$k_lendzro ;           ! SET LENGTH OF DESCRIPTOR
8            newhdrisd [isd$w_pagcnt] =                         ! SET SECTION SIZE IN PAGES
8                          (.emptydesc [1] - .emptydesc [0])/512 ;
8            newhdrisd [isd$l_flags] =                          ! COPY THE CURRENT SECTION FLAGS
9                          (.curhdrisd [isd$l_flags] or isd$m_dzro) ! AND SET DEMAND ZERO
8                          and not isd$m_crf ;                  ! CLEARING COPY ON REFERENCE
8            newhdrisd [isd$v_pfc] = 0 ;                        ! SET THE PAGE FAULT CLUSTER
8            if  .curhdrisd [isd$v_protect]                     ! IF ISECT SHOULD BE PROTECTED
8            then newhdrisd [isd$v_protect] = true ;            ! THEN DO SO
8            !
8            !      THE IMAGE SECTION DESCRIPTOR FOR THE NEW DEMAND ZERO
8            !      SECTION IS NOW SET UP. HOWEVER WE STILL HAVE TO DETERMINE
8            !      THE BASE VIRTUAL ADDRESS OF IT AND THE REMNANT OF THE ORIGINAL
8            !      IMAGE SECTION. IN ADDITION THE BUFFER ADDRESS OF THE BINARY
8            !      OF THE REMNANT MUST BE ADJUSTED.
8            !      AFTER THIS THE NEW ISD IS LINKED ONTO THE LIST OF ISD, IN
8            !      USER VIRTUAL ADDRESS ORDER.
8            !
8            if .emptydesc [0] eql .lnk$gl_curisd [isl$l_bufadr] ! IF THE EMPTY REGION IS AT
8            then
9               begin                                           ! BEGINNING OF THIS SECTION
9               previsd [isl$l_nxtisd] = .newisd ;              ! PREVIOUS ISD POINTS TO NEW
9               newhdrisd [isd$v_vpg] = .curhdrisd [isd$v_vpg] ;! AND NEW I-SECT IS AT
9               curhdrisd [isd$w_pagcnt] =                      ! THIS ONES VIRTUAL ADDRESS
9                                   .curhdrisd [isd$w_pagcnt] - ! REDUCE CURRENT I-SECT'S PAGE
9                                   .newhdrisd [isd$w_pagcnt] ; ! BY NUMBER THAT ARE EMPTY
9               newisd [isl$l_nxtisd] = .lnk$gl_curisd ;        ! ONLY PART OF SECTION
9               lnk$gl_curisd [isl$l_bufadr] = .emptydesc [1] ; ! SO NEW ISD POINTS TO
9               curhdrisd [isd$v_vpg] =                         ! CURRENT AND THE BUFFER AND
9                                   .curhdrisd [isd$v_vpg] +    ! VIRTUAL ADDRESSES OF
9                                   .newhdrisd [isd$w_pagcnt] ; ! CURRENT ARE THE PAGES
9                                                               ! BEYOND EMPTY REGION'S
9               lnk$gw_nisects = .lnk$gw_nisects + 1 ;          ! COUNT NEW IMAGE SECTION
```

```
 9                      previsd = .lnk$gl_curisd ;                             ! CURRENT ISD BECOMES PREVIOUS
 9                      end
 8                  else
 9                      begin                                                  ! EMPTY REGION IS NOT AT FRONT
 9                      newisd [isl$l_nxtisd] = .lnk$gl_curisd [isl$l_nxtisd] ; ! CURRENT POINTS TO NEW
 9                      lnk$gl_curisd [isl$l_nxtisd] = .newisd ;               ! THEN CURRENT I-SECT PAGE
 9                      curhdrisd [isd$w_pagcnt] =                             ! COUNT BECOMES NO. BEFORE
 9                              (.emptydesc [0] - .lnk$gl_curisd [isl$l_bufadr])/512 ;
 9                                                                             ! UNTOUCHED REGION
 9                      lnk$gw_imgblks = .lnk$gw_imgblks +                     ! ADD CURRENT I-SECTS
 9                                      .curhdrisd [isd$w_pagcnt] ;            ! INTO IMAGE BINARY TOTAL
 9                      newhdrisd [isd$v_vpg] = .curhdrisd [isd$v_vpg]         ! VIRTUAL ADDRESS OF
 9                                          + .curhdrisd [isd$w_pagcnt] ; ! BEYOND CURRENT
 9                      lnk$gw_nisects = .lnk$gw_nisects + 1 ;                 ! COUNT NEW IMAGE SECTION

 9                      if .emptydesc [1] gequ .lnk$gl_curisd [isl$l_bufend]       ! IF EMPTY REGION IS
 9                      then                                                       ! ALL THAT REMAINS
10                          begin                                                  !
10                          lnk$gl_curisd [isl$l_bufend] = .emptydesc [0] -1 ;     ! ADJUST END OF CURRENT
10                          previsd = .newisd ;                                    ! BUFFER AND NEW
10                          exitloop ;                                             ! BECOMES PREVIOUS
10                          end
 9                      else
10                          begin                                          ! NOW GENERATE AN ISD FOR REMNANT
10                          previsd = .newisd ;                            ! SAVING NEW AS THE PREVIOUS
10                          lnk$alloblk(isd$c_size,newisd) ;               ! ALLOCATE THE DESCRIPTOR
10                          newisd [isl$l_nxtisd] = .previsd [isl$l_nxtisd] ;       ! TO POINT
10                          previsd [isl$l_nxtisd] = .newisd ;             ! FORWARD AND POINTED TO
10                          lnk$gw_nisects = .lnk$gw_nisects + 1 ;         ! COUNT NEW IMAGE SECTION
10                          newisd [isl$l_cludsc] = .lnk$gl_curclu ;! SET CLUSTER DESCRIPTOR POINTER
10                          newisd [isl$l_bufadr] = .emptydesc [1] ;! BUFFER IS BEYOND EMPTY
10                          newisd [isl$l_bufend] =                        ! REGION TO END
10                                  .lnk$gl_curisd [isl$l_bufend] ;        ! OF REMNANT
10                          newhdrisd = newisd [isl$t_hdrisd] ;            ! NAME THE HEADER PART
10                          ch$move (.curhdrisd [isd$w_size],              ! COPY THE CURRENT ISD
10                                          curhdrisd,.newhdrisd) ;        ! TO THE NEW NE
10                          newhdrisd [isd$w_pagcnt] =
11                                          (.newisd [isl$l_bufend]        ! PAGE COUNT IS AT
11                                          -.newisd [isl$l_bufadr]        ! MOST THE REMNANT
10                                          + 511)/512 ;                   ! BUFFER SIZE
10                          newhdrisd [isd$v_vpg] =                        ! VIRTUAL ADDR IS PAGE BEYOND
10                                          .curhdrisd [isd$v_vpg]         ! THE EMPTY REGIONS
11                                          + (.newisd [isl$l_bufadr]
11                                          - .lnk$gl_curisd [isl$l_bufadr]
10                                            )/512 ;
10                          lnk$gl_curisd [isl$l_bufend] = .emptydesc [0] - 1 ; ! CURRENT BUFFER BEFORE
10                          lnk$gl_curisd = .newisd ;                      ! AND CREATE A NEW CURRENT ISD
10                          if .curhdrisd [isd$v_protect]                  ! PROTECT THE ISECT?
10                          then newhdrisd [isd$v_protect] = true ;        !
 9                          end ;                                          ! THE EMPTY REGION
 8                      end ;
 7                  end ;
 7              end
 6          until  .emptydesc [1] eql 0 ;
 5          end ;
 4      end ;
 3  end ;
```

```
;  934       1427  2      end ;
;  935       1428  2  !
;  936       1429  2  !       ALLOCATE A TABLE TO HOLD ISECT DESCRIPTOR ADDRESSES, AND FILL IT
;  937       1430  2  !       IN.  THEN SORT THE LIST INTO ASCENDING VPN ORDER.
;  938       1431  2  !
;  939       1432  2  if .lnk$gl_shradr neq 0   and  .lnk$gl_fixisd eql 0               ! CREATE FIXUP ISD IF NEEDED
;  940       1433  2  then lnk$crefixisd () ;
;  941       1434  2
;  942       1435  2  if  .lnk$gw_nisects neq 0
;  943       1436  2  then lnk$alloblk (.lnk$gw_nisects*4, isdlist)                   ! ALLOCATE SPACE FOR ISECT LIST
;  944       1437  2  else isdlist = 0 ;                                              ! (we won't fill-up this list if
;  945       1438  2                                                                  ! shareable non-based image)
;  946       1439  2  if  .lnk$gw_shriscts neq 0
;  947       1440  2  then lnk$alloblk (.lnk$gw_shriscts*4, shrisdlist)               ! AND FOR THE SHAREABLE ISECT LIST
;  948       1441  2  else shrisdlist = 0 ;
;  949       1442  2
;  950       1443  2  if  .lnk$gw_nclusters neq 0
;  951       1444  2  then lnk$alloblk (.lnk$gw_nclusters*4, shrclulst)                ! AND THE SHAREABLE CLUSTER LIST
;  952       1445  2  else shrclulst = 0 ;
;  953       1446  2
;  954       1447  2  if  .lnk$gl_shradr neq 0
;  955       1448  2  then lnk$alloblk (.lnk$gl_shradr*4, adrclulst)                   ! AND THE .ADDRESS CLUSTER REF LIST
;  956       1449  2  else adrclulst = 0 ;
;  957       1450  2
;  958       1451  2  lnk$gl_curclu = lnk$gl_clulst ;
;  959       1452  2  shrclunum     = shrisdnum = adrclunum = isdnum = 0 ;
;  960       1453  2
;  961       1454  2  while (lnk$gl_curclu = .lnk$gl_curclu [clu$l_nxtclu]) neq 0
;  962       1455  3  do  begin
;  963       1456  3      if .lnk$gl_curclu [clu$v_shrimg]                            ! IF CLUSTER IS A SHAREABLE IMAGE
;  964       1457  4      then begin
;  965       1458  4           shrclulst [.shrclunum] = .lnk$gl_curclu ;               ! SAVE CLUSTER ADDRESS IN LIST
;  966       1459  4           shrclunum = .shrclunum + 1 ;
;  967       1460  4           firstisd  = .lnk$gl_curclu [clu$l_fstisd] ;             ! GET POINTER TO FIRST ISECT DESCRIPTOR
;  968       1461  4           firstisd  = firstisd [isl$t_hdrisd] ;                   ! POINT TO PART BOUND FOR HEADER
;  969       1462  3           end ;
;  970       1463  3
;  971       1464  3      if not .lnk$gl_curclu [clu$v_shrimg]                        ! IF NOT SHAREABLE IMAGE CLUSTER
;  972       1465  3      and    .lnk$gl_curclu [clu$l_adrcnt] neq 0                  !  AND IT HAS .ADDRESS REFERENCES TO IT
;  973       1466  4      then begin
;  974       1467  4           adrclulst [.adrclunum] = .lnk$gl_curclu ;               ! SET ADDRESS INTO LIST
;  975       1468  4           adrclunum              = .adrclunum + 1 ;
;  976       1469  3           end ;
;  977       1470  3
;  978       1471  3      previsd = lnk$gl_curclu [clu$l_fstisd] ;                     ! POINT TO ISD LIST THIS CLUSTER
;  979       1472  3      while (previsd = .previsd [isl$l_nxtisd]) neq 0              ! AND RUN THRU THEM ALL
;  980       1473  4      do  begin
;  981       1474  4          bind isdhdr = previsd [isl$t_hdrisd] : block [,byte] ;  ! NAME PART BOUND FOR HEADER
;  982       1475  4          if  .isdhdr [isd$w_pagcnt] neq 0                        ! ONLY PUT ISECTS WITH MEMORY ALLOCATED
;  983       1476  5          then begin
;  984       1477  6               if not (.lnk$gl_curclu [clu$v_shrimg]               ! IF NOT A NON-BASED SHR IMG
;  985       1478  6                        and not
;  986       1479  6                       .lnk$gl_curclu [clu$v_based]
;  987       1480  6                        )
;  988       1481  6               then begin                                         ! CLUSTER IS EITHER LOCAL OR BASED
;  989       1482  6                    isdlist [.isdnum] = .previsd ;                 ! SHAREABLE IMAGE -- INSERT INTO THE
;  990       1483  6                    isdnum            = .isdnum + 1 ;              ! LIST
```

```
6                              end
6                         else begin                                     ! IT IS A  NON-BASED SHAREABLE IMAGE
6                              shrisdlist [.shrisdnum] = .previsd ;
6                              shrisdnum               = .shrisdnum + 1 ;
6                              if not .lnk$gl_curclu [clu$v_copy]       ! UNLESS TAKING A PRIVATE COPY OF THIS
7                              then  begin                              ! IMAGE
7                                    if      .isdhdr [isd$v_wrt]        ! IF THIS IS A WRITEABLE NON-CRF SECTION
7                                    and not .isdhdr [isd$v_crf]        !  (I.E. SHAREABLE WRITEABLE)
7                                    then firstisd [isd$v_wrt] = true ; !  THEN FLAG IN SECTION THAT WILL GO INTO
7                                    exitloop ;                         ! ALL DONE WITH ISECTS IN THIS
6                                    end ;                              ! PIC SHR IMG CLUSTER
5                              end ;
4                         end ;
3                    end ;
2               end ;
2
2 if  .lnk$gl_shradr neq 0
3 or (.lnk$gl_fixisd neq 0 and not .lnk$gl_ctlmsk [lnk$v_sys])
2 then
2      lnk$createfix (.shrclunum,.shrclulst,.adrclunum,.adrclulst) ;! CREATE THE FIXUP SECTION IF NECESSARY
2
2 lnk$sortisects (.isdnum, .isdlist) ;                                  ! SORT ISECTS FOR BASED AND LOCAL SECTIONS
2
2 if not .lnk$gl_ctlmsk [lnk$v_sys]                                     ! IF  NOT A SYSTEM IMAGE
2 or     .lnk$gl_ctlmsk [lnk$v_sysheadr]                                !  OR A SYSTEM IMAGE WITH A HEADER
2 then
3      begin
3      local status ;
3
3      if .isdnum    neq 0 then inchdrsize (.isdnum, .isdlist) ;            ! COUNT ISD'S INTO IMAGE HEADER
3      if .shrisdnum neq 0 then inchdrsize (.shrisdnum, .shrisdlist) ;      !  FOR SHAREABLE IMAGE ISD'S ALSO
3      inchdrsize (1, %REF (0) );                                           ! NULL ISD TERMINATES THE LIST
4      if (status = lnk$requestmem (.lnk$gw_hdrblks+1, header_desc))        ! GET SOME MEMORY FOR THE HEADER
3      then                                                                 ! (plus 1 for page aligning)
4           begin
4           header_desc [0] = (.header_desc [0] + 511) and not 511 ;            ! PAGE ALIGN THE HEADER
4           header_ptr      = ch$fill (0, ihd$k_maxlength, .header_desc [0]) ;  ! ZERO FILL FIXED PORTIO
4           header_size     = ihd$k_maxlength ;                                 ! RESET HEADER SIZE
4           ch$fill (hdr$k_fillchr, (512*.lnk$gw_hdrblks - ihd$k_maxlength), .header_ptr) ;! FILL THE REMAINDER
4           end                                                                 ! WITH FILL CHARACTER
3      else
4           begin                                                       ! BUT ON FAILURE
4           signal (.status, 0, lin$_noimgfil) ;                        ! REPORT THE ERROR
4           lnk$gl_ctlmsk [lnk$v_image] = false ;                       ! DISABLE IMAGE PRODUCTION
4           return ;                                                    ! AND QUIT
3           end ;
3
3      lnk$gw_dstblks = (.lnk$gl_dstend + 511)/512 ;                    ! COMPUTE THE NUMBER OF BLOCKS
2      end ;
2 !
2 ! OPEN THE IMAGE FILE
2 !
2 if .lnk$gl_dmtbytes neq 0
2 then dmtblks = ((.lnk$gl_dmtbytes or 511) + 1 ) / 512  ;
2
2 fileblks = .lnk$gw_hdrblks + .lnk$gw_imgblks +                        ! COMPUTE THE FILE SIZE
2            .lnk$gw_dstblks + .dmtblks ;
```

```
; 1048         1541  2
; 1049         1542  2 if  .lnk$gl_ctlmsk [lnk$v_dbg]                                    ! IF A DEBUGGABLE EXECUTABLE IMAGE
; 1050         1543  2 or  .lnk$gl_ctlmsk [lnk$v_shr]                                    ! OR A SHAREABLE IMAGE MAKE INITIAL
; 1051         1544  2 then fileblks = .fileblks + (.lnk$gw_lsymbols + 39)/20 ;           ! ALLOCATION  LARGER FOR THE SYMBOL TABLE
; 1052         1545  2
; 1053       P 1546  2 $fab_init          (fab = imagefab
; 1054       P 1547  2                    ,fac = (bro,get)
; 1055       P 1548  2                    ,mrs = 512
; 1056       P 1549  2                    ,rfm = fix
; 1057       P 1550  2                    ,alq = .fileblks
; 1058       P 1551  2                    ,fns = .lnk$gl_imgfil [fdb$w_usrnamlen]
; 1059       P 1552  2                    ,fna = .lnk$gl_imgfil [fdb$l_usrnamadr]
; 1060       P 1553  2                    ,dns = (if .lnk$gb_locnov_exe
; 1061       P 1554  2                             then %charcount ('.EXE')
; 1062       P 1555  2                             else %charcount ('SYS$DISK:[].EXE')
; 1063       P 1556  2                            )
; 1064       P 1557  2                    ,dna = (if .lnk$gb_locnov_exe
; 1065       P 1558  2                             then uplit (byte ('.EXE'))
; 1066       P 1559  2                             else uplit (byte ('SYS$DISK:[].EXE'))
; 1067       P 1560  2                            )
; 1068       P 1561  2                    ,nam = auxfnb
; 1069         1562  2                    ) ;
; 1070         1563  2
; 1071         1564  2 if  .lnk$gb_locnov_exe                                            ! USE 'OUTPUT FILE PARSING' UNLESS
; 1072         1565  2 then imagefab [fab$v_ofp] = false                                 ! /EXE WAS A LOCAL QUALIFIER
; 1073         1566  2 else imagefab [fab$v_ofp] = true ;                                ! WITH A SPECFIED VALUE
; 1074         1567  2
; 1075         1568  2 if .lnk$gl_ctlmsk [lnk$v_contig]                                  ! IF USER SAID CONTIGUOUS THEN
; 1076         1569  2 then imagefab [fab$v_ctg] = true                                  ! ATTEMPT TO DO IT, OTHERWISE
; 1077         1570  2 else imagefab [fab$v_cbt] = true ;                                ! AS CONTIGUOUS A FILE AS POSSIBLE
; 1078         1571  2
; 1079         1572  2 lnk$al_imgrab [rab$l_fab] = imagefab ;                           ! AND POINT THE RAB AT THIS FAB
; 1080         1573  2
; 1081         1574  3 if not ($getjpi (itmlst=lnk$gt_jpilst) ;
; 1082         1575  3          if .lnk$gl_filesleft leq 3
; 1083         1576  3          then lnk$closefile () ;                                  !  THEN CLOSE  A FILE
; 1084         1577  3          errorcode = $create (fab=imagefab) ;                     !  AND TRY AGAIN
; 1085         1578  3          rmsstv    = .imagefab [fab$l_stv] ;
; 1086         1579  3          .errorcode
; 1087         1580  3        )
; 1088         1581  3 or not (errorcode = $connect (rab=lnk$al_imgrab) ;                  ! CONNECT A STREAM
; 1089         1582  3          rmsstv    = .lnk$al_imgrab [rab$l_stv] ;
; 1090         1583  3          .errorcode
; 1091         1584  3        )
; 1092         1585  3 then begin                                                        ! AND IF THERE IS AN ERROR
; 1093         1586  3      signal (lin$_openout,1, lnk$filnamdsc(imagefab)              ! THEN REPORT IT
; 1094         1587  3             ,.errorcode,.rmsstv                                   ! THEN THE RMS STATUS
; 1095         1588  3             ) ;
; 1096         1589  3      lnk$gl_ctlmsk [lnk$v_image] = false ;                        ! TURN OFF IMAGE PRODUCTION
; 1097         1590  3      return ;                                                     ! THEN GIVE UP
; 1098         1591  2      end ;
; 1099         1592  2
; 1100         1593  2 ch$move (dsc$c_s_bln, lnk$filnamdsc (imagefab)                    ! SET UP REAL FILE NAME DESCRIPTOR
; 1101         1594  2         ,lnk$gl_imgfil [fdb$q_filename]
; 1102         1595  2         ) ;
; 1103         1596  2 lnk$gw_imgifi = .imagefab [fab$w_ifi] ;                           ! SAVE THE INTERNAL ID FOR CLOSE LATER
; 1104         1597  2
; 1105          1598  2 curvbn          = 1 + .lnk$gw_hdrblks ;                         ! FIRST I-SECTION STARTS AFTER HEADER
; 1106          1599  2
; 1107          1600  2 writeisects (.isdnum, .isdlist) ;                                ! WRITE OUT LOCAL AND BASED ISECTS
; 1108          1601  2 writeisects (.shrisdnum, .shrisdlist) ;                          ! WRITE OUT PIC SHAREABLE ISECTS
; 1109          1602  2
; 1110          1603  2 if .lnk$gw_hdrblks neq 0                                         ! IF IMAGE HAS A HEADER
; 1111          1604  2 then  isdtoheader (0) ;                                         ! THEN ENTER TERMINATING ISD
; 1112          1605  2
; 1113          1606  2 if  .dstbuffer [0] neq 0                                         ! IF A DST BUFFER IS ALLOCATED
; 1114          1607  2 and .lnk$gw_dstblks neq 0                                        ! AND DST WAS WRITTEN
; 1115          1608  3 then begin
; 1116          1609  3      lnk$gw_dstvbn = .curvbn ;                                   ! RECORD THE VBN OF IT
; 1117          1610  3      if not writesection (.lnk$gw_dstblks,.curvbn,.dstbuffer [0]) ! AND OUTPUT THE BLOCKS
; 1118          1611  3      then return ;
; 1119          1612  3      curvbn = .curvbn + .lnk$gw_dstblks ;                        ! SET NEXT VBN
; 1120          1613  2      end ;
; 1121          1614  2
; 1122          1615  2 if  .lnk$gl_dmtbuffer [0] neq 0                                   ! IF A DMT BUFFER IS ALLOCATED
; 1123          1616  2 and .dmtblks neq 0                                               !   AND DMT DATA WAS WRITTEN
; 1124          1617  3 then begin
; 1125          1618  3      dmtvbn = .curvbn ;                                          ! RECORD VBN
; 1126          1619  3      if not writesection (.dmtblks,.curvbn,.lnk$gl_dmtbuffer [0]) ! AND OUTPUT THE BLOCKS
; 1127          1620  3      then return ;
; 1128          1621  3      curvbn = .curvbn + .dmtblks ;                               ! UPDATE VBN
; 1129          1622  2      end ;
; 1130          1623  2
; 1131          1624  2 ! Now, if we are producing a shareable image that has no image data, then
; 1132          1625  2 !       we have to adjust the end-of-file to prepare for the correct
; 1133          1626  2 !       placement of the symbol table
; 1134          1627  2 !
; 1135          1628  2 if .lnk$gl_ctlmsk [lnk$v_shr]
; 1136          1629  2       and
; 1137          1630  2    .lnk$gw_imgblks eql 0
; 1138          1631  2 then
; 1139          1632  2     if not writesection(.lnk$gw_hdrblks, 1, .header_desc [0])
; 1140          1633  2     then
; 1141          1634  2         return;
; 1142          1635  2
; 1143          1636  2
; 1144          1637  2 if .lnk$gl_ctlmsk [lnk$v_shr]                                     ! IF PRODUCING A SHAREABLE
; 1145          1638  2 or .lnk$gl_ctlmsk [lnk$v_dbg]                                    ! OR DEBUGGABLE IMAGE
; 1146          1639  3 then begin
; 1147          1640  3      $disconnect (rab = lnk$al_imgrab) ;                         ! DISCONNECT THE RAB
; 1148          1641  3      lnk$gw_gstvbn = .curvbn ;                                   ! SET THE VBN OF THE GST
; 1149          1642  2      end ;
; 1150          1643  2
; 1151          1644  2                                                                  ! AND WILL CLOSE IT AFTER SYMBOLS
; 1152          1645  2 return ;                                                         ! AND HEADER ARE WRITTEN
; 1153          1646  1 end ;


                                                              .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                    00036     .BLKB   2
                                        45 58 45 2E 00038 P.AAE:  .ASCII  \.EXE\                          ;
45 58 45 2E 5D 5B 3A 4B 53 49 44 24 53 59 53 0003C P.AAF:  .ASCII  \SYS$DISK:[].EXE\                ;
```

```
                                                            .EXTRN  SYS$GETPTI, SYS$GETJPI
                                                            .EXTRN  SYS$CREATE, SYS$CONNECT
                                                            .EXTRN  SYS$DISCONNECT

                                                            .PSECT  $CODE$,NOWRT,2

                                          0FFC 00000        .ENTRY  LNK$FLUSHIMG, Save R2,R3,R4,R5,R6,R7,R8,R9,-; 1142
                                                                    R10,R11                                       ;
                         5E           8C   AE   9E 00002    MOVAB   -116(SP), SP                                  ;
            7E 00000000G 00                26   C1 00006    ADDL3   #38, LNK$GL_IMGFIL, -(SP)                     ; 1190
            28 00000000G 00                03   E1 0000E    BBC     #3, LNK$GL_CTLMSK, 2$                         ; 1192
            20 00000000G 00                01   E0 00016    BBS     #1, LNK$GL_CTLMSK+3, 2$                       ; 1193
                         50 00000000G      00   D0 0001E    MOVL    SYSISD, R0                                    ; 1196
                         51           18   A0   9E 00025    MOVAB   24(R0), R1                                    ;
                            00000000'      EF   D4 00029    CLRL    HEADER_SIZE                                   ; 1198
                                           50   D5 0002F    TSTL    R0                                            ; 1199
                                           08   13 00031    BEQL    1$                                            ;
               00000000'   EF         02   A1   B0 00033    MOVW    2(R1), LNK$GW_IMGBLKS                         ; 1200
                                         038D   31 0003B 1$: BRW    32$                                           ; 1192
                          FF   8F 00000000'    EF   91 0003E 2$: CMPB  LNK$GB_MATCHCTL, #255                      ; 1203
                                           14   12 00046    BNEQ    3$                                            ;
               00000000'   EF              01   90 00048    MOVB    #1, LNK$GB_MATCHCTL                           ; 1205
00000000'  EF 00000000G  00                1F   00   EF 0004F    EXTZV   #0, #31, LNK$GQ_ALOSTIM+2, LNK$GL_MATCHID ; 1206
               00000000G   00 00000000G    00   9E 0005C 3$: MOVAB  LNK$GL_CLULST, LNK$GL_CURCLU                  ; 1208
                         50 00000000G      00   D0 00067 4$: MOVL   LNK$GL_CURCLU, R0                             ; 1209
               00000000G   00              60   D0 0006E    MOVL    (R0), LNK$GL_CURCLU                           ;
                                           C4   13 00075    BEQL    1$                                            ;
                         50 00000000G      00   D0 00077    MOVL    LNK$GL_CURCLU, R0                             ; 1211
                                           51   D4 0007E    CLRL    R1                                            ;
               00000000G   00              50   D1 00080    CMPL    R0, LNK$GL_LSTCLSTR                           ;
                                           02   12 00087    BNEQ    5$                                            ;
                                           51   D6 00089    INCL    R1                                            ;
        52 00000000G  00         01        02   EF 0008B 5$: EXTZV  #2, #1, LNK$GL_CTLMSK, R2                     ; 1212
                   08   AE       51        52   CB 00094    BICL3   R2, R1, LASTCLUSTER                           ;
                                 57   18   A0   9E 00099    MOVAB   24(R0), PREVISD                               ; 1213
               00000000'   EF              67   D0 0009D 6$: MOVL   (PREVISD), LNK$GL_CURISD                      ; 1214
                                           C1   13 000A4    BEQL    4$                                            ;
                         50 00000000'      EF   D0 000A6    MOVL    LNK$GL_CURISD, R0                             ; 1217
                         51           18   A0   9E 000AD    MOVAB   24(R0), R1                                    ;
               00000000G   00              50   D1 000B1    CMPL    R0, LNK$GL_FIXISD                             ; 1219
                                           0A   12 000B8    BNEQ    7$                                            ;
               00000000'   EF         02   A1   A0 000BA    ADDW2   2(R1), LNK$GW_IMGBLKS                         ; 1221
                                           0E   11 000C2    BRB     8$                                            ; 1222
                         54           02   A1   9E 000C4 7$: MOVAB  2(R1), R4                                     ; 1225
                                           64   B5 000C8    TSTW    (R4)                                          ;
                                           09   12 000CA    BNEQ    9$                                            ;
                            00000000G      00   B7 000CC    DECW    LNK$GW_NISECTS                                ; 1227
                                         00FA   31 000D2 8$: BRW    22$                                           ; 1228
                            00000000G      00   95 000D5 9$: TSTB   LNK$GL_CTLMSK+2                               ; 1232
                                           11   19 000DB    BLSS    10$                                           ;
                         52 00000000G      00   D0 000DD    MOVL    LNK$GL_CURCLU, R2                             ; 1233
                                      58   A2   95 000E4    TSTB    88(R2)                                        ;
                                           05   19 000E7    BLSS    10$                                           ;
                  04         0A   A1       01   E1 000E9    BBC     #1, 10(R1), 11$                               ; 1234
                             0A   A1       04   88 000EE 10$: BISB2 #4, 10(R1)                                    ; 1235
                         05           08   AE   E9 000F2 11$: BLBC LASTCLUSTER, 12$                               ; 1237
```

```
          08  A1        80    8F 88 000F6        BISB2    #128, 8(R1)                     ; 1238
              52        08    A1 9E 000FB 12$:   MOVAB    8(R1), R2                       ; 1240
6F            62              02 E0 000FF        BBS      #2, (R2), 19$                   ;
                        08    A0 D5 00103        TSTL     8(R0)                           ; 1241
                              6A 12 00106        BNEQ     19$                             ;
1D            62              03 E1 00108        BBC      #3, (R2), 14$                   ; 1242
              1D              62 E8 0010C        BLBS     (R2), 15$                       ; 1243
07 00000000G  00              02 E1 0010F        BBC      #2, LNK$GL_CTLMSK, 13$          ; 1244
              0B 00000000G    00 E9 00117        BLBC     LNK$GL_UNSUPPORTED, 14$         ; 1246
              61              0C B0 0011E 13$:   MOVW     #12, (R1)                       ; 1249
              62              04 88 00121        BISB2    #4, (R2)                        ; 1250
              62              02 8A 00124        BICB2    #2, (R2)                        ; 1251
                              49 11 00127        BRB      19$                             ; 1242
              22              62 E9 00129 14$:   BLBC     (R2), 17$                       ; 1253
              53 00000000G    00 D0 0012C 15$:   MOVL     LNK$GL_CURCLU, R3               ; 1255
0E        58  A3              04 E1 00133        BBC      #4, 88(R3), 16$                 ;
0A            62              12 E0 00138        BBS      #18, (R2), 16$                  ; 1256
          08  A0 00000000G    00 7D 0013C        MOVQ     LNK$GL_GSBUFDSC+4, 8(R0)        ; 1258
                              2C 11 00144        BRB      19$                             ; 1255
                        0C    A1 D4 00146 16$:   CLRL     12(R1)                          ; 1262
                        08    A0 7C 00149        CLRQ     8(R0)                           ; 1263
                              24 11 0014C        BRB      19$                             ; 1253
                        08    A0 9F 0014E 17$:   PUSHAB   8(R0)                           ; 1270
              7E              64 3C 00151        MOVZWL   (R4), -(SP)                     ;
   00000000G  00              02 FB 00154        CALLS    #2, LNK$REQUESTMEM              ;
              14              50 E8 0015B        BLBS     STATUS, 19$                     ; 1272
                 00000000G    8F DD 0015E 18$:   PUSHL    #LIN$_NOIMGFIL                  ; 1274
                              7E D4 00164        CLRL     -(SP)                           ;
                              50 DD 00166        PUSHL    STATUS                          ;
   00000000G  00              03 FB 00168        CALLS    #3, LIB$SIGNAL                  ;
                            05C7 31 0016F        BRW      69$                             ; 1275
              50 00000000'    EF D0 00172 19$:   MOVL     LNK$GL_CURISD, R0               ; 1280
                        08    A0 D5 00179        TSTL     8(R0)                           ;
                              21 13 0017C        BEQL     21$                             ;
   00000000G  00              64 B1 0017E        CMPW     (R4), LNK$GW_DZROMIN            ; 1282
                              11 1F 00185        BLSSU    20$                             ;
0D            62              03 E1 00187        BBC      #3, (R2), 20$                   ; 1283
   00000000G  00 00000000G    00 B1 0018B        CMPW     LNK$GW_NISECTS, LNK$GW_MISECTS  ; 1284
                              07 1F 00196        BLSSU    21$                             ;
   00000000'  EF              64 A0 00198 20$:   ADDW2    (R4), LNK$GW_IMGBLKS            ; 1286
2C            62              02 E0 0019F 21$:   BBS      #2, (R2), 22$                   ; 1288
28            62              03 E1 001A3        BBC      #3, (R2), 22$                   ; 1289
              25              62 E8 001A7        BLBS     (R2), 22$                       ; 1290
   00000000G  00              64 B1 001AA        CMPW     (R4), LNK$GW_DZROMIN            ; 1291
                              1C 1F 001B1        BLSSU    22$                             ;
   00000000G  00 00000000G    00 B1 001B3        CMPW     LNK$GW_NISECTS, LNK$GW_MISECTS  ; 1292
                              0F 1E 001BE        BGEQU    22$                             ;
0D 00000000G  00              02 E1 001C0        BBC      #2, LNK$GL_CTLMSK, 23$          ; 1293
              06 00000000G    00 E8 001C8        BLBS     LNK$GL_UNSUPPORTED, 23$         ; 1295
              57              50 D0 001CF 22$:   MOVL     R0, PREVISD                     ; 1297
                            FEC8 31 001D2        BRW      6$                              ;
58 00000000'  EF              18 C1 001D5 23$:   ADDL3    #24, LNK$GL_CURISD, R8          ; 1300
              7E 00000000G    00 3C 001DD        MOVZWL   LNK$GW_DZROMIN, -(SP)           ; 1308
                              01 DD 001E4        PUSHL    #1                              ;
                              7E D4 001E6        CLRL     -(SP)                           ;
                        7C    AE 9F 001E8        PUSHAB   EMPTYDESC                       ;
7E 00000000'  EF              08 C1 001EB        ADDL3    #8, LNK$GL_CURISD, -(SP)        ;
```

```
                       00000000G  00            05 FB 001F3        CALLS   #5, SYS$GETPTI                               ;
                              04  AE            50 D0 001FA        MOVL    R0, ERRORCODE                                ;
                                  32        04  AE E8 001FE        BLBS    ERRORCODE, 24$                               ;
                                     00000000G  8F DD 00202        PUSHL   #LIN$_NOIMGFIL                               ; 1310
                                                7E D4 00208        CLRL    -(SP)                                        ;
                                            0C  AE DD 0020A        PUSHL   ERRORCODE                                    ; 1315
                                  7E        02  A8 3C 0020D        MOVZWL  2(R8), -(SP)                                 ; 1314
      7E       04   A8            17            00 EF 00211        EXTZV   #0, #23, 4(R8), -(SP)                        ; 1313
                                  50 00000000'  EF D0 00217        MOVL    LNK$GL_CURISD, R0                            ; 1312
                                  7E        08  A0 7D 0021E        MOVQ    8(R0), -(SP)                                 ; 1311
                                                04 DD 00222        PUSHL   #4                                           ; 1310
                                     00000000G  8F DD 00224        PUSHL   #LIN$_GETPTIERR                              ;
                       00000000G  00            09 FB 0022A        CALLS   #9, LIB$SIGNAL                               ;
                                              0505 31 00231        BRW     69$                                          ; 1317
                                            74  AE D6 00234 24$:   INCL    EMPTYDESC+4                                  ; 1320
                                                0D 13 00237        BEQL    25$                                          ;
                       00000000G  00 00000000G  00 B1 00239        CMPW    LNK$GW_NISECTS, LNK$GW_MISECTS               ; 1321
                                                15 1F 00244        BLSSU   26$                                          ;
                                  57 00000000'  EF D0 00246 25$:   MOVL    LNK$GL_CURISD, PREVISD                       ; 1323
                       00000000'  EF        02  A8 A0 0024D        ADDW2   2(R8), LNK$GW_IMGBLKS                        ; 1324
                                            74  AE D4 00255        CLRL    EMPTYDESC+4                                  ; 1325
                                              0165 31 00258        BRW     30$                                          ; 1320
                                            0C  AE 9F 0025B 26$:   PUSHAB  NEWISD                                       ; 1331
                                  7E        58  8F 9A 0025E        MOVZBL  #88, -(SP)                                   ;
                       00000000G  00            02 FB 00262        CALLS   #2, LNK$ALLOBLK                              ;
                                  59        0C  AE D0 00269        MOVL    NEWISD, R9                                   ; 1332
                              10  A9 00000000G  00 D0 0026D        MOVL    LNK$GL_CURCLU, 16(R9)                        ;
                                            08  A9 7C 00275        CLRQ    8(R9)                                        ; 1333
                                  56        18  A9 9E 00278        MOVAB   24(R9), NEWHDRISD                            ; 1335
                                  66            0C B0 0027C        MOVW    #12, (NEWHDRISD)                             ; 1336
                    50        74  AE        70  AE C3 0027F        SUBL3   EMPTYDESC, EMPTYDESC+4, R0                   ; 1338
                    51            50  00000200  8F C7 00285        DIVL3   #512, R0, R1                                 ;
                              02  A6            51 B0 0028D        MOVW    R1, 2(NEWHDRISD)                             ;
                    50        08  A8            02 CB 00291        BICL3   #2, 8(R8), R0                                ; 1341
               08   A6            50            04 C9 00296        BISL3   #4, R0, 8(NEWHDRISD)                         ;
                                            07  A6 94 0029B        CLRB    7(NEWHDRISD)                                 ; 1342
                    04        0A  A8            02 E1 0029E        BBC     #2, 10(R8), 27$                              ; 1343
                              0A  A6            04 88 002A3        BISB2   #4, 10(NEWHDRISD)                            ; 1344
                                  50 00000000'  EF D0 002A7 27$:   MOVL    LNK$GL_CURISD, R0                            ; 1355
                              08  A0        70  AE D1 002AE        CMPL    EMPTYDESC, 8(R0)                             ;
                                                36 12 002B3        BNEQ    28$                                          ;
                                  67            59 D0 002B5        MOVL    R9, (PREVISD)                                ; 1358
04    A6            17            00        04  A8 F0 002B8        INSV    4(R8), #0, #23, 4(NEWHDRISD)                 ; 1359
                              02  A8        02  A6 A2 002BF        SUBW2   2(NEWHDRISD), 2(R8)                          ; 1362
                                  69            50 D0 002C4        MOVL    R0, (R9)                                     ; 1363
                              08  A0        74  AE D0 002C7        MOVL    EMPTYDESC+4, 8(R0)                           ; 1364
      51       04   A8            17            00 EF 002CC        EXTZV   #0, #23, 4(R8), R1                           ; 1367
                                  52        02  A6 3C 002D2        MOVZWL  2(NEWHDRISD), R2                             ;
                                  51            52 C0 002D6        ADDL2   R2, R1                                       ;
04    A8            17            00            51 F0 002D9        INSV    R1, #0, #23, 4(R8)                           ;
                                     00000000G  00 B6 002DF        INCW    LNK$GW_NISECTS                               ; 1369
                                  57            50 D0 002E5        MOVL    R0, PREVISD                                  ; 1370
                                              00D5 31 002E8        BRW     30$                                          ; 1355
                                  69            60 D0 002EB 28$:   MOVL    (R0), (R9)                                   ; 1374
                                  60            59 D0 002EE        MOVL    R9, (R0)                                     ; 1375
                    51        70  AE        08  A0 C3 002F1        SUBL3   8(R0), EMPTYDESC, R1                         ; 1377
                    52            51  00000200  8F C7 002F7        DIVL3   #512, R1, R2                                 ;
```

```
                                   02   A8            52 B0 002FF          MOVW    R2, 2(R8)                            ;
                            00000000'   EF        02  A8 A0 00303          ADDW2   2(R8), LNK$GW_IMGBLKS                ; 1380
            51       04   A8            17            00 EF 0030B          EXTZV   #0, #23, 4(R8), R1                   ; 1382
                                        52        02  A8 3C 00311          MOVZWL  2(R8), R2                            ;
                                        51            52 C0 00315          ADDL2   R2, R1                               ;
       04   A6            17            00            51 F0 00318          INSV    R1, #0, #23, 4(NEWHDRISD)            ;
                                           00000000G  00 B6 0031E          INCW    LNK$GW_NISECTS                       ; 1383
                          5B       70   AE            01 C3 00324          SUBL3   #1, EMPTYDESC, R11                   ; 1388
                                        57            59 D0 00329          MOVL    R9, PREVISD                          ; 1389
                                   0C   A0        74  AE D1 0032C          CMPL    EMPTYDESC+4, 12(R0)                  ; 1385
                                                      07 1F 00331          BLSSU   29$                                  ;
                                   0C   A0            5B D0 00333          MOVL    R11, 12(R0)                          ; 1388
                                                    FD63 31 00337          BRW     6$                                   ; 1387
                                                  0C  AE 9F 0033A 29$:     PUSHAB  NEWISD                               ; 1395
                                        7E        58  8F 9A 0033D          MOVZBL  #88, -(SP)                           ;
                            00000000G   00            02 FB 00341          CALLS   #2, LNK$ALLOBLK                      ;
                                        59        0C  AE D0 00348          MOVL    NEWISD, R9                           ; 1396
                                        69            67 D0 0034C          MOVL    (PREVISD), (R9)                      ;
                                        67            59 D0 0034F          MOVL    R9, (PREVISD)                        ; 1397
                                           00000000G  00 B6 00352          INCW    LNK$GW_NISECTS                       ; 1398
                                   10   A9 00000000G  00 D0 00358          MOVL    LNK$GL_CURCLU, 16(R9)                ; 1399
                                   08   A9        74  AE D0 00360          MOVL    EMPTYDESC+4, 8(R9)                   ; 1400
                                        5A 00000000'  EF D0 00365          MOVL    LNK$GL_CURISD, R10                   ; 1402
                                   0C   A9        0C  AA D0 0036C          MOVL    12(R10), 12(R9)                      ;
                                        56        18  A9 9E 00371          MOVAB   24(R9), NEWHDRISD                    ; 1403
                          66            68            68 28 00375          MOVC3   (R8), (R8), (NEWHDRISD)              ; 1405
                          50       0C   A9        08  A9 C3 00379          SUBL3   8(R9), 12(R9), R0                    ; 1408
                                        50      01FF  C0 9E 0037F          MOVAB   511(R0), R0                          ; 1407
                          51            50  00000200  8F C7 00384          DIVL3   #512, R0, R1                         ; 1409
                                   02   A6            51 B0 0038C          MOVW    R1, 2(NEWHDRISD)                     ;
                          50       08   A9        08  AA C3 00390          SUBL3   8(R10), 8(R9), R0                    ; 1413
                                        50  00000200  8F C6 00396          DIVL2   #512, R0                             ; 1414
            51       04   A8            17            00 EF 0039D          EXTZV   #0, #23, 4(R8), R1                   ;
                                        51            50 C0 003A3          ADDL2   R0, R1                               ;
       04   A6            17            00            51 F0 003A6          INSV    R1, #0, #23, 4(NEWHDRISD)            ;
                                   0C   AA            5B D0 003AC          MOVL    R11, 12(R10)                         ; 1415
                            00000000'   EF            59 D0 003B0          MOVL    R9, LNK$GL_CURISD                    ; 1416
                          04       0A   A8            02 E1 003B7          BBC     #2, 10(R8), 30$                      ; 1417
                                   0A   A6            04 88 003BC          BISB2   #4, 10(NEWHDRISD)                    ; 1418
                                                  74  AE D5 003C0 30$:     TSTL    EMPTYDESC+4                          ; 1423
                                                      03 12 003C3          BNEQ    31$                                  ;
                                                    FCD5 31 003C5          BRW     6$                                   ;
                                                    FE0A 31 003C8 31$:     BRW     23$                                  ;
                                           00000000G  00 D5 003CB 32$:     TSTL    LNK$GL_SHRADR                        ; 1432
                                                      0F 13 003D1          BEQL    33$                                  ;
                                           00000000G  00 D5 003D3          TSTL    LNK$GL_FIXISD                        ;
                                                      07 12 003D9          BNEQ    33$                                  ;
                            00000000G   00            00 FB 003DB          CALLS   #0, LNK$CREFIXISD                    ; 1433
                                        50 00000000G  00 3C 003E2 33$:     MOVZWL  LNK$GW_NISECTS, R0                   ; 1435
                                                      10 13 003E9          BEQL    34$                                  ;
                                                  10  AE 9F 003EB          PUSHAB  ISDLIST                              ; 1436
                          7E            50            02 78 003EE          ASHL    #2, R0, -(SP)                        ;
                            00000000G   00            02 FB 003F2          CALLS   #2, LNK$ALLOBLK                      ;
                                                      03 11 003F9          BRB     35$                                  ;
                                                  10  AE D4 003FB 34$:     CLRL    ISDLIST                              ; 1437
                                        50 00000000G  00 3C 003FE 35$:     MOVZWL  LNK$GW_SHRISCTS, R0                  ; 1439
                                                      10 13 00405          BEQL    36$                                  ;
```

```
                        14  AE  9F 00407         PUSHAB  SHRISDLIST                              ; 1440
7E            50            02  78 0040A         ASHL    #2, R0, -(SP)                           ;
   00000000G  00            02  FB 0040E         CALLS   #2, LNK$ALLOBLK                         ;
                            03  11 00415         BRB     37$                                     ;
                        14  AE  D4 00417 36$:    CLRL    SHRISDLIST                              ; 1441
              50 00000000'  EF  3C 0041A 37$:    MOVZWL  LNK$GW_NCLUSTERS, R0                    ; 1443
                            10  13 00421         BEQL    38$                                     ;
                        18  AE  9F 00423         PUSHAB  SHRCLULST                               ; 1444
7E            50            02  78 00426         ASHL    #2, R0, -(SP)                           ;
   00000000G  00            02  FB 0042A         CALLS   #2, LNK$ALLOBLK                         ;
                            03  11 00431         BRB     39$                                     ;
                        18  AE  D4 00433 38$:    CLRL    SHRCLULST                               ; 1445
              50 00000000G  00  D0 00436 39$:    MOVL    LNK$GL_SHRADR, R0                       ; 1447
                            10  13 0043D         BEQL    40$                                     ;
                        1C  AE  9F 0043F         PUSHAB  ADRCLULST                               ; 1448
7E            50            02  78 00442         ASHL    #2, R0, -(SP)                           ;
   00000000G  00            02  FB 00446         CALLS   #2, LNK$ALLOBLK                         ;
                            03  11 0044D         BRB     41$                                     ;
                        1C  AE  D4 0044F 40$:    CLRL    ADRCLULST                               ; 1449
   00000000G  00 00000000G  00  9E 00452 41$:    MOVAB   LNK$GL_CLULST, LNK$GL_CURCLU            ; 1451
                            52  7C 0045D         CLRQ    ADRCLUNUM                               ; 1452
                            59  7C 0045F         CLRQ    ISDNUM                                  ;
              50 00000000G  00  D0 00461         MOVL    LNK$GL_CURCLU, R0                       ; 1454
   00000000G  00            60  D0 00468 42$:    MOVL    (R0), LNK$GL_CURCLU                     ;
                            6B  13 0046F         BEQL    48$                                     ;
              50 00000000G  00  D0 00471         MOVL    LNK$GL_CURCLU, R0                       ; 1456
13       58   A0            02  E1 00478         BBC     #2, 88(R0), 43$                         ;
         18 BE43            50  D0 0047D         MOVL    R0, @SHRCLULST[SHRCLUNUM]               ; 1458
                            53  D6 00482         INCL    SHRCLUNUM                               ; 1459
              54        18  A0  D0 00484         MOVL    24(R0), FIRSTISD                        ; 1460
              54            18  C0 00488         ADDL2   #24, FIRSTISD                           ; 1461
0C       58   A0            02  E0 0048B         BBS     #2, 88(R0), 44$                         ; 1464
                        34  A0  D5 00490 43$:    TSTL    52(R0)                                  ; 1465
                            07  13 00493         BEQL    44$                                     ;
         1C BE42            50  D0 00495         MOVL    R0, @ADRCLULST[ADRCLUNUM]               ; 1467
                            52  D6 0049A         INCL    ADRCLUNUM                               ; 1468
              57        18  A0  9E 0049C 44$:    MOVAB   24(R0), PREVISD                         ; 1471
              57            67  D0 004A0 45$:    MOVL    (PREVISD), PREVISD                      ; 1472
                            C3  13 004A3         BEQL    42$                                     ;
              51        18  A7  9E 004A5         MOVAB   24(PREVISD), R1                         ; 1474
                        02  A1  B5 004A9         TSTW    2(R1)                                   ; 1475
                            F2  13 004AC         BEQL    45$                                     ;
04       58   A0            02  E1 004AE         BBC     #2, 88(R0), 46$                         ; 1477
              09        58  A0  E9 004B3         BLBC    88(R0), 47$                             ; 1479
         10 BE49            57  D0 004B7 46$:    MOVL    PREVISD, @ISDLIST[ISDNUM]               ; 1482
                            59  D6 004BC         INCL    ISDNUM                                  ; 1483
                            E0  11 004BE         BRB     45$                                     ; 1477
         14 BE4A            57  D0 004C0 47$:    MOVL    PREVISD, @SHRISDLIST[SHRISDNUM]         ; 1486
                            5A  D6 004C5         INCL    SHRISDNUM                               ; 1487
D4       58   A0            04  E0 004C7         BBS     #4, 88(R0), 45$                         ; 1488
97       08   A1            03  E1 004CC         BBC     #3, 8(R1), 42$                          ; 1490
92       08   A1            01  E0 004D1         BBS     #1, 8(R1), 42$                          ; 1491
         08   A4            08  88 004D6         BISB2   #8, 8(FIRSTISD)                         ; 1492
                            8C  11 004DA         BRB     42$                                     ; 1489
                 00000000G  00  D5 004DC 48$:    TSTL    LNK$GL_SHRADR                           ; 1500
                            10  12 004E2         BNEQ    49$                                     ;
                 00000000G  00  D5 004E4         TSTL    LNK$GL_FIXISD                           ; 1501
```

```
                                                     19  13 004EA         BEQL    50$                                        ;
                         11 00000000G  00            03  E0 004EC         BBS     #3, LNK$GL_CTLMSK, 50$                     ;
                                                1C   AE  DD 004F4 49$:    PUSHL   ADRCLULST                                  ; 1503
                                                     52  DD 004F7         PUSHL   ADRCLUNUM                                  ;
                                                20   AE  DD 004F9         PUSHL   SHRCLULST                                  ;
                                                     53  DD 004FC         PUSHL   SHRCLUNUM                                  ;
                            00000000V  EF            04  FB 004FE         CALLS   #4, LNK$CREATEFIX                          ;
                                                10   AE  DD 00505 50$:    PUSHL   ISDLIST                                    ; 1505
                                                     59  DD 00508         PUSHL   ISDNUM                                     ;
                            00000000G  00            02  FB 0050A         CALLS   #2, LNK$SORTISECTS                         ;
                         0B 00000000G  00            03  E1 00511         BBC     #3, LNK$GL_CTLMSK, 51$                     ; 1507
                         03 00000000G  00            01  E0 00519         BBS     #1, LNK$GL_CTLMSK+3, 51$                   ; 1508
                                                   00B4  31 00521         BRW     55$                                        ;
                                                     59  D5 00524 51$:    TSTL    ISDNUM                                     ; 1513
                                                     0C  13 00526         BEQL    52$                                        ;
                                                10   AE  DD 00528         PUSHL   ISDLIST                                    ;
                                                     59  DD 0052B         PUSHL   ISDNUM                                     ;
                            00000000V  EF            02  FB 0052D         CALLS   #2, INCHDRSIZE                             ;
                                                     5A  D5 00534 52$:    TSTL    SHRISDNUM                                  ; 1514
                                                     0C  13 00536         BEQL    53$                                        ;
                                                14   AE  DD 00538         PUSHL   SHRISDLIST                                 ;
                                                     5A  DD 0053B         PUSHL   SHRISDNUM                                  ;
                            00000000V  EF            02  FB 0053D         CALLS   #2, INCHDRSIZE                             ;
                                                08   AE  D4 00544 53$:    CLRL    8(SP)                                      ; 1515
                                                08   AE  9F 00547         PUSHAB  8(SP)                                      ;
                                                     01  DD 0054A         PUSHL   #1                                         ;
                            00000000V  EF            02  FB 0054C         CALLS   #2, INCHDRSIZE                             ;
                                          00000000'  EF  9F 00553         PUSHAB  HEADER_DESC                                ; 1516
                                       7E 00000000'  EF  3C 00559         MOVZWL  LNK$GW_HDRBLKS, -(SP)                      ;
                                                     6E  D6 00560         INCL    (SP)                                       ;
                            00000000G  00            02  FB 00562         CALLS   #2, LNK$REQUESTMEM                         ;
                                       03            50  E8 00569         BLBS    STATUS, 54$                                ;
                                                   FBEF  31 0056C         BRW     18$                                        ;
                         50 00000000'  EF 000001FF   8F  C1 0056F 54$:    ADDL3   #511, HEADER_DESC, R0                      ; 1519
              00000000'  EF            50 000001FF   8F  CB 0057B         BICL3   #511, R0, HEADER_DESC                      ;
    00A8   8F            00            6E            00  2C 00587         MOVC5   #0, (SP), #0, #168, @HEADER_DESC           ; 1520
                                          00000000'  FF     0058E                                                            ;
                            00000000'  EF            53  D0 00593         MOVL    R3, HEADER_PTR                             ;
                            00000000'  EF       A8   8F  9A 0059A         MOVZBL  #168, HEADER_SIZE                          ; 1521
                                       50 00000000'  EF  3C 005A2         MOVZWL  LNK$GW_HDRBLKS, R0                         ; 1522
                         50            50            09  78 005A9         ASHL    #9, R0, R0                                 ;
                                       50     FF58   C0  9E 005AD         MOVAB   -168(R0), R0                               ;
           50       FF   8F            6E            00  2C 005B2         MOVC5   #0, (SP), #255, R0, @HEADER_PTR            ;
                                          00000000'  FF     005B8                                                            ;
                         50 00000000'  EF 000001FF   8F  C1 005BD         ADDL3   #511, LNK$GL_DSTEND, R0                    ; 1531
                         51            50 00000200   8F  C7 005C9         DIVL3   #512, R0, R1                               ;
                            00000000'  EF            51  B0 005D1         MOVW    R1, LNK$GW_DSTBLKS                         ;
                                       50 00000000G  00  D0 005D8 55$:    MOVL    LNK$GL_DMTBYTES, R0                        ; 1536
                                                     13  13 005DF         BEQL    56$                                        ;
                                       50     01FF   8F  A8 005E1         BISW2   #511, R0                                   ; 1537
                                                     50  D6 005E6         INCL    R0                                         ;
              00000000'  EF            50 00000200   8F  C7 005E8         DIVL3   #512, R0, DMTBLKS                          ;
                                       50 00000000'  EF  3C 005F4 56$:    MOVZWL  LNK$GW_HDRBLKS, R0                         ; 1539
                                       51 00000000'  EF  3C 005FB         MOVZWL  LNK$GW_IMGBLKS, R1                         ;
                                       50            51  C0 00602         ADDL2   R1, R0                                     ;
                                       51 00000000'  EF  3C 00605         MOVZWL  LNK$GW_DSTBLKS, R1                         ; 1540
                                       50            51  C0 0060C         ADDL2   R1, R0                                     ; 1539
              56           50 00000000' EF C1 0060F       ADDL3    DMTBLKS, R0, FILEBLKS            ; 1540
              08 00000000G 00           06 E0 00617       BBS      #6, LNK$GL_CTLMSK, 57$           ; 1542
              10 00000000G 00           02 E1 0061F       BBC      #2, LNK$GL_CTLMSK, 58$           ; 1543
                           50 00000000G 00 3C 00627 57$:  MOVZWL   LNK$GW_LSYMBOLS, R0              ; 1544
                           50           27 C0 0062E       ADDL2    #39, R0                          ;
                           50           14 C6 00631       DIVL2    #20, R0                          ;
                           56           50 C0 00634       ADDL2    R0, FILEBLKS                     ;
0050    8F    00           6E           00 2C 00637 58$:  MOVC5    #0, (SP), #0, #80, $RMS_PTR      ; 1562
                                     20 AE    0063E                                                 ;
                        20 AE      5003 8F B0 00640       MOVW     #20483, $RMS_PTR                 ;
                        30 AE           56 D0 00646       MOVL     FILEBLKS, $RMS_PTR+16            ;
                        36 AE        42 8F 90 0064A       MOVB     #66, $RMS_PTR+22                 ;
                        3F AE           01 90 0064F       MOVB     #1, $RMS_PTR+31                  ;
                        48 AE           6E D0 00653       MOVL     (SP), $RMS_PTR+40                ;
                           50 00000000G 00 D0 00657       MOVL     LNK$GL_IMGFIL, R0                ;
                        4C AE        10 A0 D0 0065E       MOVL     16(R0), $RMS_PTR+44              ;
                           52 00000000G 00 9A 00663       MOVZBL   LNK$GB_LOCNOV_EXE, R2            ;
                           09           52 E9 0066A       BLBC     R2, 59$                          ;
                           51 00000000' EF 9E 0066D       MOVAB    P.AAE, R1                        ;
                                        07 11 00674       BRB      60$                              ;
                           51 00000000' EF 9E 00676 59$:  MOVAB    P.AAF, R1                        ;
                        50 AE           51 D0 0067D 60$:  MOVL     R1, $RMS_PTR+48                  ;
                        54 AE        0C A0 90 00681       MOVB     12(R0), $RMS_PTR+52              ;
                           05           52 E9 00686       BLBC     R2, 61$                          ;
                           50           04 D0 00689       MOVL     #4, R0                           ;
                                        03 11 0068C       BRB      62$                              ;
                           50           0F D0 0068E 61$:  MOVL     #15, R0                          ;
                        55 AE           50 90 00691 62$:  MOVB     R0, $RMS_PTR+53                  ;
                        56 AE      0200 8F B0 00695       MOVW     #512, $RMS_PTR+54                ;
                        06              52 E9 0069B       BLBC     R2, 63$                          ; 1564
                        27 AE           20 8A 0069E       BICB2    #32, IMAGEFAB+7                  ; 1565
                                        04 11 006A2       BRB      64$                              ;
                        27 AE           20 88 006A4 63$:  BISB2    #32, IMAGEFAB+7                  ; 1566
              06 00000000G 00           03 E1 006A8 64$:  BBC      #3, LNK$GL_CTLMSK+2, 65$         ; 1568
                        26 AE           10 88 006B0       BISB2    #16, IMAGEFAB+6                  ; 1569
                                        04 11 006B4       BRB      66$                              ;
                        26 AE           20 88 006B6 65$:  BISB2    #32, IMAGEFAB+6                  ; 1570
                 00000000' EF        20 AE 9E 006BA 66$:  MOVAB    IMAGEFAB, LNK$AL_IMGRAB+60       ; 1572
                                        7E 7C 006C2       CLRQ     -(SP)                            ; 1574
                                        7E D4 006C4       CLRL     -(SP)                            ;
                              00000000G 00 9F 006C6       PUSHAB   LNK$GT_JPILST                    ;
                                        7E 7C 006CC       CLRQ     -(SP)                            ;
                                        7E D4 006CE       CLRL     -(SP)                            ;
                 00000000G 00           07 FB 006D0       CALLS    #7, SYS$GETJPI                   ;
                           03 00000000G 00 D1 006D7       CMPL     LNK$GL_FILESLEFT, #3             ; 1575
                                        07 14 006DE       BGTR     67$                              ;
                 00000000G 00           00 FB 006E0       CALLS    #0, LNK$CLOSEFILE                ; 1576
                                     20 AE 9F 006E7 67$:  PUSHAB   IMAGEFAB                         ; 1577
                 00000000G 00           01 FB 006EA       CALLS    #1, SYS$CREATE                   ;
                        04 AE           50 D0 006F1       MOVL     R0, ERRORCODE                    ;
                           52        2C AE D0 006F5       MOVL     IMAGEFAB+12, RMSSTV              ; 1578
                           1C        04 AE E9 006F9       BLBC     ERRORCODE, 68$                   ; 1579
                              00000000' EF 9F 006FD       PUSHAB   LNK$AL_IMGRAB                    ; 1581
                 00000000G 00           01 FB 00703       CALLS    #1, SYS$CONNECT                  ;
                        04 AE           50 D0 0070A       MOVL     R0, ERRORCODE                    ;
                           52 00000000' EF D0 0070E       MOVL     LNK$AL_IMGRAB+12, RMSSTV         ; 1582
                           28        04 AE E8 00715       BLBS     ERRORCODE, 70$                   ; 1583
```

```
                                52  DD 00719  68$:   PUSHL   RMSSTV                       ; 1587
                             08 AE  DD 0071B         PUSHL   ERRORCODE                    ;
                             28 AE  9F 0071E         PUSHAB  IMAGEFAB                     ; 1586
        00000000G  00           01  FB 00721         CALLS   #1, LNK$FILNAMDSC            ;
                                50  DD 00728         PUSHL   R0                           ;
                                01  DD 0072A         PUSHL   #1                           ;
                      00000000G 8F  DD 0072C         PUSHL   #LIN$_OPENOUT                ;
        00000000G  00           05  FB 00732         CALLS   #5, LIB$SIGNAL               ;
        00000000G  00           01  8A 00739  69$:   BICB2   #1, LNK$GL_CTLMSK            ; 1589
                                    04 00740         RET                                  ; 1585
                             20 AE  9F 00741  70$:   PUSHAB  IMAGEFAB                     ; 1593
        00000000G  00           01  FB 00744         CALLS   #1, LNK$FILNAMDSC            ;
                   51 00000000G 00  D0 0074B         MOVL    LNK$GL_IMGFIL, R1            ; 1594
14   A1            60           08  28 00752         MOVC3   #8, (R0), 20(R1)             ;
        00000000'  EF        22 AE  B0 00757         MOVW    IMAGEFAB+2, LNK$GW_IMGIFI    ; 1596
        00000000'  EF 00000000' EF  3C 0075F         MOVZWL  LNK$GW_HDRBLKS, CURVBN       ; 1598
                      00000000' EF  D6 0076A         INCL    CURVBN                       ;
                             10 AE  DD 00770         PUSHL   ISDLIST                      ; 1600
                                59  DD 00773         PUSHL   ISDNUM                       ;
        00000000V  EF           02  FB 00775         CALLS   #2, WRITEISECTS              ;
                             14 AE  DD 0077C         PUSHL   SHRISDLIST                   ; 1601
                                5A  DD 0077F         PUSHL   SHRISDNUM                    ;
        00000000V  EF           02  FB 00781         CALLS   #2, WRITEISECTS              ;
                      00000000' EF  B5 00788         TSTW    LNK$GW_HDRBLKS               ; 1603
                                09  13 0078E         BEQL    71$                          ;
                                7E  D4 00790         CLRL    -(SP)                        ; 1604
        00000000V  EF           01  FB 00792         CALLS   #1, ISDTOHEADER              ;
                   50 00000000' EF  D0 00799  71$:   MOVL    DSTBUFFER, R0                ; 1606
                                3A  13 007A0         BEQL    72$                          ;
                      00000000' EF  B5 007A2         TSTW    LNK$GW_DSTBLKS               ; 1607
                                32  13 007A8         BEQL    72$                          ;
        00000000'  EF 00000000' EF  B0 007AA         MOVW    CURVBN, LNK$GW_DSTVBN        ; 1609
                                50  DD 007B5         PUSHL   R0                           ; 1610
                      00000000' EF  DD 007B7         PUSHL   CURVBN                       ;
                   7E 00000000' EF  3C 007BD         MOVZWL  LNK$GW_DSTBLKS, -(SP)        ;
        00000000V  EF           03  FB 007C4         CALLS   #3, WRITESECTION             ;
                   73           50  E9 007CB         BLBC    R0, 74$                      ;
                   50 00000000' EF  3C 007CE         MOVZWL  LNK$GW_DSTBLKS, R0           ; 1612
        00000000'  EF           50  C0 007D5         ADDL2   R0, CURVBN                   ;
                   50 00000000G 00  D0 007DC  72$:   MOVL    LNK$GL_DMTBUFFER, R0         ; 1615
                                36  13 007E3         BEQL    73$                          ;
                      00000000' EF  D5 007E5         TSTL    DMTBLKS                      ; 1616
                                2E  13 007EB         BEQL    73$                          ;
        00000000'  EF 00000000' EF  D0 007ED         MOVL    CURVBN, DMTVBN               ; 1618
                                50  DD 007F8         PUSHL   R0                           ; 1619
                      00000000' EF  DD 007FA         PUSHL   CURVBN                       ;
                      00000000' EF  DD 00800         PUSHL   DMTBLKS                      ;
        00000000V  EF           03  FB 00806         CALLS   #3, WRITESECTION             ;
                   5C           50  E9 0080D         BLBC    R0, 78$                      ;
        00000000'  EF 00000000' EF  C0 00810         ADDL2   DMTBLKS, CURVBN              ; 1621
     29 00000000G  00           02  E1 0081B  73$:   BBC     #2, LNK$GL_CTLMSK, 76$       ; 1628
                      00000000' EF  B5 00823         TSTW    LNK$GW_IMGBLKS               ; 1630
                                19  12 00829         BNEQ    75$                          ;
                      00000000' EF  DD 0082B         PUSHL   HEADER_DESC                  ; 1632
                                01  DD 00831         PUSHL   #1                           ;
                   7E 00000000' EF  3C 00833         MOVZWL  LNK$GW_HDRBLKS, -(SP)        ;
        00000000V  EF           03  FB 0083A         CALLS   #3, WRITESECTION             ;
```

```
                          28                  50  E9 00841 74$:    BLBC    R0, 78$                                     ;
             08 00000000G  00                  02  E0 00844 75$:    BBS     #2, LNK$GL_CTLMSK, 77$                      ; 1637
             18 00000000G  00                  06  E1 0084C 76$:    BBC     #6, LNK$GL_CTLMSK, 78$                      ; 1638
                                 00000000'     EF  9F 00854 77$:    PUSHAB  LNK$AL_IMGRAB                               ; 1640
                00000000G  00                  01  FB 0085A         CALLS   #1, SYS$DISCONNECT                          ;
                00000000'  EF 00000000'        EF  B0 00861         MOVW    CURVBN, LNK$GW_GSTVBN                       ; 1641
                                                   04 0086C 78$:    RET                                                 ; 1646

; Routine Size:  2157 bytes,    Routine Base:  $CODE$ + 0458
```

```
1 routine inchdrsize (numisds, isd_list) : novalue =
1 !
1 !       THIS ROUTINE RUNS THROUGH THE ISD'S IN THE LIST AND
1 !       ADDS THEIR HEADER SIZE CONTRIBUTION UP.  THIS ROUTINE
1 !       MUST BE CALLED AFTER THE ISD'S HAVE BEEN SORTED.
1 !
1 !       THIS ROUTINE IS CALLED WITH isdlist POINTING TO A
1 !       LONGWORD OF ZERO WITH numisds=1 TO TERMINATE THE
1 !       ENTIRE SEQUENCE OF ISD's.
1 !
2 begin
2 map
2       isd_list            : ref vector [,long];
2 own
2       bytes_in_block      : initial (first_block_size) ;
2 local
2       isd_size,
2       isd                 : ref block [,byte],
2       hdrisd              : ref block [,byte];
2
2 incru i from 0 to .numisds-1                                ! PROCESS LIST OF ISD's
3 do begin
3    isd    = .isd_list [.i];
3    hdrisd =  isd [isl$t_hdrisd];
3
4    isd_size =    (if .isd eql 0                             ! IF ZERO THEN TERMINATING ISD
4                   then  hdr$k_minfill
4                   else .hdrisd [isd$w_size]
3                  );
4    if .bytes_in_block lssu (.header_size + .isd_size)      ! IF THIS ISD WILL NOT FIT WITHIN
4    then begin                                              ! THE CURRENT HEADER BLOCK
4         lnk$gw_hdrblks   = .lnk$gw_hdrblks + 1;            ! INCREMENT THE BLOCK COUNT
4         header_size      =  0;                             ! INITIALIZE HEADER SIZE THIS BLOCK
4         bytes_in_block   =  subseq_block_size;             ! RESET USEABLE BYTES
3         end;
3
3    header_size = .header_size + .isd_size;                 ! ADD TO THE CURRENT BLOCK SIZE
2    end;
2
2 if .isd_list [0] eql 0                                     ! IF TERMINATING ISD, ACCOUNT FOR
2 then lnk$gw_hdrblks = .lnk$gw_hdrblks + 1 ;                ! THIS, THE LAST, HEADER BLOCK
2
2 return;
1 end;



                                        .PSECT  $OWN$,NOEXE,2

                 000001FC  00070 BYTES_IN_BLOCK:
                                        .LONG   508                                    ;



                                        .PSECT  $CODE$,NOWRT,2

                        00FC 00000 INCHDRSIZE:
```

```
                                                         .WORD   Save R2,R3,R4,R5,R6,R7                        ; 1647
                     57 00000000'  EF   9E 00002         MOVAB   LNK$GW_HDRBLKS, R7                            ;
                     56 00000000'  EF   9E 00009         MOVAB   HEADER_SIZE, R6                               ;
      55        04   AC            01   C3 00010         SUBL3   #1, NUMISDS, R5                               ; 1667
                                   53   D4 00015         CLRL    I                                             ; 1669
                                   2E   11 00017         BRB     5$                                            ;
                     50       08 BC43   D0 00019 1$:     MOVL    @ISD_LIST[I], ISD                             ;
                     51       18   A0   9E 0001E         MOVAB   24(R0), HDRISD                                ; 1670
                                   50   D5 00022         TSTL    ISD                                           ; 1672
                                   05   12 00024         BNEQ    2$                                            ;
                     54            02   D0 00026         MOVL    #2, ISD_SIZE                                  ;
                                   03   11 00029         BRB     3$                                            ;
                     54            61   3C 0002B 2$:     MOVZWL  (HDRISD), ISD_SIZE                            ; 1674
      52             66            54   C1 0002E 3$:     ADDL3   ISD_SIZE, HEADER_SIZE, R2                     ; 1676
                     52       04   A6   D1 00032         CMPL    BYTES_IN_BLOCK, R2                            ;
                                   0A   1E 00036         BGEQU   4$                                            ;
                                   67   B6 00038         INCW    LNK$GW_HDRBLKS                                ; 1678
                                   66   D4 0003A         CLRL    HEADER_SIZE                                   ; 1679
                04   A6     01FC   8F   3C 0003C         MOVZWL  #508, BYTES_IN_BLOCK                          ; 1680
                     66            54   C0 00042 4$:     ADDL2   ISD_SIZE, HEADER_SIZE                         ; 1683
                                   53   D6 00045         INCL    I                                             ; 1667
                     55            53   D1 00047 5$:     CMPL    I, R5                                         ;
                                   CD   1B 0004A         BLEQU   1$                                            ;
                              08   BC   D5 0004C         TSTL    @ISD_LIST                                     ; 1686
                                   02   12 0004F         BNEQ    6$                                            ;
                                   67   B6 00051         INCW    LNK$GW_HDRBLKS                                ; 1687
                                        04 00053 6$:     RET                                                   ; 1690

; Routine Size:  84 bytes,    Routine Base:  $CODE$ + 0CC5
; 1200        1691  1 routine isdtoheader (isd) : novalue =
; 1201        1692  1 !
; 1202        1693  1 !         THIS ROUTINE COPIES IMAGE SECTION DESCRIPTORS TO THE
; 1203        1694  1 !         IMAGE HEADER AND ENSURES THAT THE UNUSED PART OF EACH
; 1204        1695  1 !         BLOCK IS -1 FILLED (I.E. VALUE OF HDR$K_FILLCHR). IF THE
; 1205        1696  1 !         ARGUMENT IS 0, IT IS ASSUMED THAT THIS IS THE END OF THE
; 1206        1697  1 !         ISD LIST FOR THE HEADER.
; 1207        1698  1 !
; 1208        1699  2 begin
; 1209        1700  2 map
; 1210        1701  2           isd : ref block [,byte];                          ! ARGUMENT IS A POINTER
; 1211        1702  2 own
; 1212        1703  2           bytes_in_block  : initial (first_block_size) ;    ! FIRST BLOCK HAS 2 BYTES RESERVED
; 1213        1704  2 local
; 1214        1705  2           isd_size;
; 1215        1706  2
; 1216        1707  3 isd_size    =   (if .isd eql 0                              ! IF TERMINATING (NULL) ISD
; 1217        1708  3                  then  hdr$k_minfill                        ! USE FILL SIZE
; 1218        1709  3                  else .isd [isd$w_size]                     ! ELSE USE ACTUAL SIZE
; 1219        1710  2                 );
; 1220        1711  3 if .bytes_in_block lssu (.header_size + .isd_size)          ! IF THIS ISD WILL NOT FIT
; 1221        1712  3 then begin                                                  ! DESCRIPTOR IN THE CURRENT BLOCK
; 1222        1713  3       header_ptr      =  (.header_ptr + 511) and not 511;   ! THEN ADVANCE TO NEXT BLOCK
; 1223        1714  3       header_size     =  0;                                 ! RESET SIZE OF THIS BLOCK
; 1224        1715  3       bytes_in_block =  subseq_block_size;                  ! RESET USEABLE BYTES
; 1225        1716  2       end;
; 1226        1717  2
; 1227        1718  2 header_size = .header_size + .isd_size;                     ! UPDATE HEADER SIZE
; 1228        1719  3 header_ptr  = ( if  .isd neq 0                              ! IF THIS ISN'T THE END OF IT
; 1229        1720  3                 then ch$move (.isd_size, .isd, .header_ptr) ! JUST COPY IT IN
; 1230        1721  4                 else begin                                  ! OTHERWIZE...
; 1231        1722  4                         if .lnk$gl_ctlmsk [lnk$v_cli]       ! IF IMAGE IS CLI,
; 1232        1723  4                         then (.header_desc [0] + 510)<0,16> = 3;  ! THEN TELL IMAGE ACTIVATOR HERE
; 1233        1724  4                         ch$fill (0, .isd_size, .header_ptr) ! INSERT MINIMUM ISD OF 0 (END OF LIST)
; 1234        1725  4                         end
; 1235        1726  2               );
; 1236        1727  2 return;
; 1237        1728  1 end;


                                                   .PSECT  $OWN$,NOEXE,2

                         000001FC  00074 BYTES_IN_BLOCK:
                                                   .LONG   508                                     ;


                                                   .PSECT  $CODE$,NOWRT,2

                             00FC 00000 ISDTOHEADER:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7                  ; 1691
               57 00000000'   EF   9E 00002        MOVAB   HEADER_PTR, R7                          ;
               51        04   AC   D0 00009        MOVL    ISD, R1                                 ; 1707
                              05   12 0000D        BNEQ    1$                                      ;
               56             02   D0 0000F        MOVL    #2, ISD_SIZE                            ;
                              03   11 00012        BRB     2$                                      ;
```

```
                                        56            61  3C 00014 1$:     MOVZWL  (R1), ISD_SIZE                                 ; 1709
                         50             56       04   A7  C1 00017 2$:     ADDL3   HEADER_SIZE, ISD_SIZE, R0                      ; 1711
                                        50       0C   A7  D1 0001C         CMPL    BYTES_IN_BLOCK, R0                             ;
                                                      19  1E 00020         BGEQU   3$                                             ;
                         50             67 000001FF   8F  C1 00022         ADDL3   #511, HEADER_PTR, R0                           ; 1713
                         67             50 000001FF   8F  CB 0002A         BICL3   #511, R0, HEADER_PTR                           ;
                                                 04   A7  D4 00032         CLRL    HEADER_SIZE                                    ; 1714
                                  0C    A7     01FC   8F  3C 00035         MOVZWL  #508, BYTES_IN_BLOCK                           ; 1715
                                  04    A7            56  C0 0003B 3$:     ADDL2   ISD_SIZE, HEADER_SIZE                          ; 1718
                                                      51  D5 0003F         TSTL    R1                                             ; 1719
                                                      07  13 00041         BEQL    4$                                             ;
                    00   B7             61            56  28 00043         MOVC3   ISD_SIZE, (R1), @HEADER_PTR                    ; 1720
                                                      18  11 00048         BRB     6$                                             ;
                         09 00000000G   00            05  E1 0004A 4$:     BBC     #5, LNK$GL_CTLMSK+3, 5$                        ; 1722
                                        50       F8   A7  D0 00052         MOVL    HEADER_DESC, R0                                ; 1723
                                 01FE   C0            03  B0 00056         MOVW    #3, 510(R0)                                    ;
           56            00             6E            00  2C 0005B 5$:     MOVC5   #0, (SP), #0, ISD_SIZE, @HEADER_PTR            ; 1724
                                                 00   B7     00060                                                                ;
                                        67            53  D0 00062 6$:     MOVL    R3, HEADER_PTR                                 ; 1719
                                                          04 00065         RET                                                    ; 1728

; Routine Size:  102 bytes,    Routine Base:  $CODE$ + 0D19
; 1239          1729  1 routine writeisects (numisects, isd_list) =
; 1240          1730  1 !
; 1241          1731  1 ! THIS ROUTINE WRITES OUT THE SPECIFIED NUMBER OF IMAGE SECTIONS TO
; 1242          1732  1 ! THE IMAGE FILE.  ISDLIST IS A LIST OF THE ADDRESSES OF THE ISD'S
; 1243          1733  1 ! TO WRITE
; 1244          1734  1 !
; 1245          1735  2 begin
; 1246          1736  2 map
; 1247          1737  2     isd_list : ref vector [,long];
; 1248          1738  2
; 1249          1739  2 if  .numisects neq 0
; 1250          1740  2 then incru isdnum from 0 to .numisects- 1                       ! LOOP THRU ALL IN LIST
; 1251          1741  3 do begin
; 1252          1742  3    lnk$gl_curisd = .isd_list [.isdnum];                         ! POINT TO ISECT DESCRIPTOR
; 1253          1743  4         begin
; 1254          1744  4         bind curhdrisd = lnk$gl_curisd [isl$t_hdrisd]           ! NAME PART FOR HEADER
; 1255          1745  4                                               : block [,byte];
; 1256          1746  4         lnk$gl_curclu = .lnk$gl_curisd [isl$l_cludsc];          ! SET CLUSTER DESCRIPTOR ADDRESS
; 1257          1747  4
; 1258          1748  4         if  .curhdrisd [isd$v_gbl]                              ! IF A GLOBAL SECTION
; 1259          1749  4         and .lnk$gl_curisd [isl$l_bufadr] neq 0                 ! WITH A BUFFER ALLOCATED
; 1260          1750  4         then readsection ();                                    ! GO READ IT FROM ITS FILE
; 1261          1751  4
; 1262          1752  4         if not .curhdrisd [isd$v_dzro]                          ! IF NOT A DEMAND ZERO SECTION
; 1263          1753  4         and    .lnk$gl_curisd [isl$l_bufadr] neq 0              ! AND A BUFFER IS ALLOCATED
; 1264          1754  4         then curhdrisd [isd$l_vbn] = .curvbn;                   ! INSERT THE VBN OF BINARY
; 1265          1755  4
; 1266          1756  5         if  (if   .lnk$gl_ctlmsk [lnk$v_sys]                    ! IF A SYSTEM IMAGE WITH A HEADER
; 1267          1757  5              then .lnk$gl_ctlmsk [lnk$v_sysheadr]               ! ...
; 1268          1758  5              else true                                          ! OR A REGULAR IAMGE
; 1269          1759  5             )
; 1270          1760  5 !?      if  .lnk$gw_hdrblks neq 0                               ! IF IMAGE HAS A HEADER
; 1271          1761  4         and .curhdrisd [isd$w_pagcnt] neq 0                     ! AND SECTION IS NON ZERO IN LENGTH
; 1272          1762  4         then isdtoheader (curhdrisd);                           ! INSERT ISD IN HEADER
; 1273          1763  4
; 1274          1764  4         if not .curhdrisd [isd$v_dzro]                          ! IF NOT A DEMAND ZERO SECTION
; 1275          1765  4         and    .lnk$gl_curisd [isl$l_bufadr] neq 0              ! AND A BUFFER IS ALLOCATED
; 1276          1766  5         then begin
; 1277          1767  5              if not writesection (.curhdrisd [isd$w_pagcnt]     ! THEN WRITE IT OUT
; 1278          1768  5                                  ,.curvbn
; 1279          1769  5                                  ,.lnk$gl_curisd [isl$l_bufadr] ! TO THE FILE
; 1280          1770  5                                  )
; 1281          1771  5              then return false;                                 ! GIVING UP ON ERROR
; 1282          1772  5              curvbn = .curvbn + .curhdrisd [isd$w_pagcnt];      ! UPDATE TO NEXT AVAILABLE VBN
; 1283          1773  4              end;
; 1284          1774  3         end;
; 1285          1775  2     end;
; 1286          1776  2 return true
; 1287          1777  1 end;




                                                     007C 00000 WRITEISECTS:
                                                                  .WORD   Save R2,R3,R4,R5,R6                        ; 1729
                                   56 00000000'  EF  9E 00002     MOVAB   CURVBN, R6                                 ;
```

```
                                        55 00000000'  EF 9E 00009         MOVAB   LNK$GL_CURISD, R5                           ;
                                                 04   AC D5 00010         TSTL    NUMISECTS                                   ; 1739
                                                      03 12 00013         BNEQ    1$                                          ;
                                                    008E 31 00015         BRW     9$                                          ;
                          54        04  AC            01 C3 00018 1$:     SUBL3   #1, NUMISECTS, R4                           ; 1740
                                                      53 D4 0001D         CLRL    ISDNUM                                      ; 1742
                                                      7D 11 0001F         BRB     8$                                          ;
                                        65       08 BC43 D0 00021 2$:     MOVL    @ISD_LIST[ISDNUM], LNK$GL_CURISD            ;
                                        50            65 D0 00026         MOVL    LNK$GL_CURISD, R0                           ; 1744
                                        52       18   A0 9E 00029         MOVAB   24(R0), R2                                  ;
                             00000000G  00       10   A0 D0 0002D         MOVL    16(R0), LNK$GL_CURCLU                       ; 1746
                                        0C       08   A2 E9 00035         BLBC    8(R2), 3$                                   ; 1748
                                                 08   A0 D5 00039         TSTL    8(R0)                                       ; 1749
                                                      07 13 0003C         BEQL    3$                                          ;
                             00000000V  EF            00 FB 0003E         CALLS   #0, READSECTION                             ; 1750
                          0C        08  A2            02 E0 00045 3$:     BBS     #2, 8(R2), 4$                               ; 1752
                                        50            65 D0 0004A         MOVL    LNK$GL_CURISD, R0                           ; 1753
                                                 08   A0 D5 0004D         TSTL    8(R0)                                       ;
                                                      04 13 00050         BEQL    4$                                          ;
                                    0C  A2            66 D0 00052         MOVL    CURVBN, 12(R2)                              ; 1754
                          08 00000000G  00            03 E1 00056 4$:     BBC     #3, LNK$GL_CTLMSK, 5$                       ; 1756
                          0C 00000000G  00            01 E1 0005E         BBC     #1, LNK$GL_CTLMSK+3, 6$                     ; 1757
                                                 02   A2 B5 00066 5$:     TSTW    2(R2)                                       ; 1761
                                                      07 13 00069         BEQL    6$                                          ;
                                                      52 DD 0006B         PUSHL   R2                                          ; 1762
                                  FF28  CF            01 FB 0006D         CALLS   #1, ISDTOHEADER                             ;
                          25        08  A2            02 E0 00072 6$:     BBS     #2, 8(R2), 7$                               ; 1764
                                        50            65 D0 00077         MOVL    LNK$GL_CURISD, R0                           ; 1765
                                                 08   A0 D5 0007A         TSTL    8(R0)                                       ;
                                                      1D 13 0007D         BEQL    7$                                          ;
                                        50            65 D0 0007F         MOVL    LNK$GL_CURISD, R0                           ; 1769
                                                 08   A0 DD 00082         PUSHL   8(R0)                                       ;
                                                      66 DD 00085         PUSHL   CURVBN                                      ; 1768
                                        7E       02   A2 3C 00087         MOVZWL  2(R2), -(SP)                                ; 1767
                             00000000V  EF            03 FB 0008B         CALLS   #3, WRITESECTION                            ;
                                        15            50 E9 00092         BLBC    R0, 10$                                     ;
                                        50       02   A2 3C 00095         MOVZWL  2(R2), R0                                   ; 1772
                                        66            50 C0 00099         ADDL2   R0, CURVBN                                  ;
                                                      53 D6 0009C 7$:     INCL    ISDNUM                                      ; 1740
                                        54            53 D1 0009E 8$:     CMPL    ISDNUM, R4                                  ;
                                                      03 1A 000A1         BGTRU   9$                                          ;
                                                    FF7B 31 000A3         BRW     2$                                          ;
                                        50            01 D0 000A6 9$:     MOVL    #1, R0                                      ; 1776
                                                         04 000A9         RET                                                 ;
                                                      50 D4 000AA 10$:    CLRL    R0                                          ; 1777
                                                         04 000AC         RET                                                 ;

; Routine Size:  173 bytes,    Routine Base:  $CODE$ + 0D7F
; 1289          1778  1 global routine lnk$getfixsize =
; 1290          1779  2 begin
; 1291          1780  2 !
; 1292          1781  2 ! THIS ROOUTINE COMPUTES AND RETURNS AS A VALUE THE SIZE (IN PAGES)
; 1293          1782  2 ! REQUIRED FOR THE FIXUP SECTION
; 1294          1783  2 !
; 1295          1784  2
; 1296          1785  6 return ((((.lnk$gl_shrimgs*2)+(.lnk$gl_shrsyms+1)                    ! NUMBER OF SHR IMGS REFERENCED
; 1297          1786  5                 +.lnk$gl_spcgfix                                      ! PLUS NUMBER OF SPECIAL G^ FIXUPS
; 1298          1787  4                 +(.lnk$gl_shradr*2)+(.lnk$gl_adrefs+1))*4)            ! PLUS THE SIZE FOR THE .ADDR FIXUPS
; 1299          1788  3                 +(.lnk$gl_shrclstrs+1)*shl$c_length                   ! PLUS THE SIZE OF THE SHL LIST
; 1300          1789  4                 +(.lnk$gl_shlextra*shl$c_length)                      ! PLUS ANY EXTRA SHL'S REQUESTED
; 1301          1790  3                 +(.lnk$gl_rprotisd+1)*icp$c_length+4                  ! PLUS THE SIZE OF THE FIXUP INFO
; 1302          1791  2                 +iaf$c_length+511)/512                                ! PLUS THE FIXED PART, ALL ROUNDED UP TO NEX
; 1303          1792  1 end;



                                    0000 00000     .ENTRY  LNK$GETFIXSIZE, Save nothing                        ; 1778
                51 00000000G   00    D0 00002     MOVL    LNK$GL_SHRIMGS, R1                                   ; 1785
                50 00000000G   00    D0 00009     MOVL    LNK$GL_SHRSYMS, R0                                   ;
                50           6041    3E 00010     MOVAW   (R0)[R1], R0                                         ;
                51 00000000G   00    D0 00014     MOVL    LNK$GL_SPCGFIX, R1                                   ; 1786
                51             50    C0 0001B     ADDL2   R0, R1                                               ;
                50 00000000G   00    D0 0001E     MOVL    LNK$GL_SHRADR, R0                                    ; 1787
                51           6140    3E 00025     MOVAW   (R1)[R0], R1                                         ;
                51 00000000G   00    C0 00029     ADDL2   LNK$GL_ADREFS, R1                                    ;
   50 00000000' EF             06    78 00030     ASHL    #6, LNK$GL_SHRCLSTRS, R0                             ; 1788
                50           6041    DE 00038     MOVAL   (R0)[R1], R0                                         ;
   51 00000000G 00             06    78 0003C     ASHL    #6, LNK$GL_SHLEXTRA, R1                              ; 1789
                51             50    C0 00044     ADDL2   R0, R1                                               ;
                50 00000000'   EF    D0 00047     MOVL    LNK$GL_RPROTISD, R0                                  ; 1790
                50      0293 C140    7E 0004E     MOVAQ   659(R1)[R0], R0                                      ; 1785
                50 00000200    8F    C6 00054     DIVL2   #512, R0                                             ; 1791
                                     04 0005B     RET                                                          ; 1792

; Routine Size:  92 bytes,    Routine Base:  $CODE$ + 0E2C
; 1305          1793  1 routine storespcgfix (node,travinfo) =
; 1306          1794  2 begin
; 1307          1795  2 !
; 1308          1796  2 ! THIS ROUTINE IS CALLED BY THE TRAVERSE_TREE ROUTINE FOR EACH
; 1309          1797  2 ! ELEMENT IN THE TREE.
; 1310          1798  2 !
; 1311          1799  2 ! INPUTS:
; 1312          1800  2 !
; 1313          1801  2 !       NODE            ADDRESS OF CURRENT NODE
; 1314          1802  2 !       TRAVINFO        ADDRESS OF INFO SET UP BY LNK$CREATEFIX
; 1315          1803  2 !
; 1316          1804  2 map
; 1317          1805  2     node : ref block[,byte],
; 1318          1806  2     travinfo : ref vector[,long];
; 1319          1807  2
; 1320          1808  2 bind
; 1321          1809  2     bufloc = .travinfo[0] + (.node[spcr$l_fixadr] - .travinfo[1]),
; 1322          1810  2     storeaddr = bufloc : vector[,long];
; 1323          1811  2
; 1324          1812  2 storeaddr[0] = .node[spcr$l_offset];
; 1325          1813  2 return true
; 1326          1814  1 end;



                                                   0004 00000 STORESPCGFIX:
                                                                .WORD    Save R2                                   ; 1793
                        50          08    AC    D0 00002        MOVL     TRAVINFO, R0                              ; 1809
                        52                60    D0 00006        MOVL     (R0), R2                                  ;
                        51          04    AC    D0 00009        MOVL     NODE, R1                                  ;
          50      0E    A1          04    A0    C3 0000D        SUBL3    4(R0), 14(R1), R0                         ;
                                        6042    9F 00013        PUSHAB   (R0)[R2]                                  ; 1812
                        9E          0A    A1    D0 00016        MOVL     10(R1), @(SP)+                            ;
                        50                01    D0 0001A        MOVL     #1, R0                                    ; 1813
                                                04 0001D        RET                                                ; 1814

; Routine Size:  30 bytes,    Routine Base:  $CODE$ + 0E88
```

```
1 routine lnk$createfix (shrclunum,clusterlist,adrclunum,adrclulst) =
1 !
1 ! THIS ROUTINE ALLOCATES BUFFER SPACE AND FILLS IN THE
1 ! FIXUP IMAGE SECTION FOR COMMUNICATION WITH THE IMAGE
1 ! ACTIVATOR.
1 !
1 ! INPUTS:
1 !
1 !          SHRCLUNUM        NUMBER OF ITEMS IN CLUSTERLIST
1 !          CLUSTERLIST      ADDRESS OF LIST OF CLUSTER DESCRIPTOR ADDRESSES OF SHAREABLE IMAGES
1 !          ADRCLUNUM        NUMBER OF ITEMS IN ADRCLULST
1 !          ADRCLULST        ADDRESS OF LIST OF CLUSTER DESCRIPTOR ADDRESSES OF NON-SHR IMAGE CLUSTERS
1 !                            REFERENCED VIA .ADDRESS
1 !
2 begin
2 map
2     clusterlist           : ref vector [,long],
2     adrclulst             : ref vector [,long] ;
2 local
2     status,
2     clusterindex,
2     fixsize,
2     fixbuf                : ref block [,byte],
2     symbolblock           : ref block [,byte],
2     adrblock              : ref vector [,long],
2     shrnamlist            : ref block [,byte],
2     shrimgindex,
2     curisd                : ref block [,byte],
2     curhdrisd             : ref block [,byte],
2     spcgfixvec            : ref vector [,long],
2     travinfo              : vector [2,long],                            ! INFO FOR ACTION ROUTINE OF TREE TRAVERSAL
2     reprotvec             : ref block [,byte],
2     extadrvec             : ref vector [,long],
2     extindex,
2     fixisdhdr             : ref block [,byte],
2     dotadrvec             : ref vector [,long],
2     dotadrindex ;
2
2 fixisdhdr = lnk$gl_fixisd [isl$t_hdrisd] ;                             ! POINT TO PART BOUND FOR IMAGE HEADER
2
2 if .lnk$gl_ctlmsk [lnk$v_protect]                                     ! IF /PROTECT
2 then fixisdhdr [isd$v_protect] = true ;                               ! THEN PROTECT SECTION
2
2 fixisdhdr [isd$v_fixupvec] = true ;                                   ! DECLARE ISD TO BE FIXUP VECTOR SECTION
2 fixsize                    = lnk$getfixsize () ;                      ! GET SIZE OF FIXUP SECTION NEEDED
2 fixisdhdr [isd$w_pagcnt]   = .fixsize ;                               ! SET NUMBER OF PAGES INTO ISD
2 status = lnk$requestmem (.fixsize, lnk$gl_fixisd [isl$l_bufadr]);     ! ALLOCATE SPACE FOR SECTION
2 if not .status
3 then begin                                                            ! BUT IF IT FAILS, ISSUE AN ERROR MESSAGE
3     signal (.status, 0, lin$_noimgfil) ;
3     lnk$gl_ctlmsk [lnk$v_image] = false ;
3     return true
2     end ;
2 !
2 ! ALLOCATE VIRTUAL MEMORY FOR FIXUP SECTION.  THIS IS DONE TO ENSURE THAT
2 ! WE DID NOT BASE THE WORLD DURING PASS 2 AND PUT A SHAREABLE IMAGE WHERE
2 ! WE THOUGHT THE FIXUP SECTION WAS GOING TO GO
```

```
2 !
2 if not .lnk$gl_fixisd [isl$v_memalo]
3 then begin
3      if not lnk$allovirmem (.fixisdhdr [isd$v_vpn]^9,.fixisdhdr [isd$w_pagcnt])
4      then begin
4           signal (lin$_confixmem,1,lnk$gl_imgfil [fdb$q_filename],lin$_noimgfil) ;
4           lnk$gl_ctlmsk [lnk$v_image] = false ;
4           return true
3           end ;
3      lnk$gl_fixisd [isl$v_memalo] = true ;
2      end ;
2
2 fixbuf        = .lnk$gl_fixisd [isl$l_bufadr] ;
2 travinfo [0] = .fixbuf ;                                      ! SET INFO FOR TREE TRAVERSAL
2 travinfo [1] = .fixisdhdr [isd$v_vpn] ^ 9 ;
2
2 fixbuf [iaf$l_iaflink]   =  0 ;                              ! ZERO IMAGE ACTIVATOR LINK
2 fixbuf [iaf$w_size]      =  iaf$c_length ;                   ! SET SIZE OF FIXED PART OF FIXUP ISD HEADER
2 fixbuf [iaf$v_shr]       = .lnk$gl_ctlmsk [lnk$v_shr] ;      ! SET FLAGS
2 fixbuf [iaf$l_shlextra]  = .lnk$gl_shlextra ;                ! SET COUNT OF EXTRA SHL SLOTS
2 fixbuf [iaf$l_shrimgcnt] = .lnk$gl_shrclstrs + 1 ;           ! SET COUNT OF SHAREABLE IMAGES
2                                                              !  (+1 FOR CURRENT IMAGE)
2
2 extadrvec  = .fixbuf + iaf$c_length ;                        ! SET ADDRESS OF START OF G^ VECTOR
2 spcgfixvec = .extadrvec +                                    ! SET ADDRESS OF SPECIAL G^ FIXUP VECTOR
2              ((.lnk$gl_shrimgs*2)+(.lnk$gl_shrsyms+1))*4 ;
2
2 if .lnk$gl_spcgfix neq 0                                     ! IF THERE ARE SPECIAL G^ FIXUPS
2 then reprotvec = (.lnk$gl_lastgadr - .travinfo [1]) + .fixbuf !  THEN SET REPROT VEC AFTER THAT
2 else reprotvec = .spcgfixvec ;                               !  OTHERWISE RIGHT ON IT
2
2 shrnamlist = .reprotvec + (.lnk$gl_rprotisd+1)*icp$c_length + 4 ;  ! SET ADDRESS OF SHR IMG NAME LIST
2 dotadrvec  = .shrnamlist +                                         ! SET ADDRESS OF .ADDR VECTOR
2              (.lnk$gl_shrclstrs+1+.lnk$gl_shlextra)*shl$c_length ;
2
2 !
2 ! SET OFFSETS TO THE SEGMENTS INTO THE FIXUP ISECT HEADER
2 !
2 fixbuf [iaf$l_chgprtoff] = .reprotvec - .fixbuf ;            ! SET OFFSET TO CHANGE PROTECTION DATA
2
2 if .lnk$gl_shrsyms neq 0                                     ! IF THERE ARE G^ FIXUPS
2 then fixbuf [iaf$l_g_fixoff] = .extadrvec - .fixbuf          ! THEN CALCULATE OFFSET
2 else fixbuf [iaf$l_g_fixoff] = 0 ;                           ! OTHERWISE GUARANTEE A 0
2
2 if .lnk$gl_adrefs neq 0                                      ! IF THERE ARE .ADDR FIXUPS
2 then fixbuf [iaf$l_dotadroff] = .dotadrvec - .fixbuf         ! THEN CALCULATE OFFSET
2 else fixbuf [iaf$l_dotadroff] = 0 ;                          ! OTHERWISE GUARANTEE A 0
2
2 fixbuf [iaf$l_shlstoff] = .shrnamlist - .fixbuf ;            ! CALCULATE OFFSET TO SHR IMG NAME LIST
2 dotadrindex = shrimgindex =  extindex = 0 ;                  ! INIT INDICES INTO THE VECTORS
2
2 !
2 ! SET REPROTECTION DATA FOR FIXUP SECTION
2 !
2 .reprotvec = .lnk$gl_rprotisd + 1 ;
2 reprotvec  = .reprotvec + 4 ;
2 reprotvec [icp$l_baseva] = .fixisdhdr [isd$v_vpn]^9 - .lnk$gl_minva ;   ! SET BASE ADDRESS
```

```
; 1442          1929  2 reprotvec [icp$w_npages] = .fixsize ;
; 1443          1930  2 reprotvec [icp$w_newprt] = prt$c_urew ;
; 1444          1931  2 reprotvec = .reprotvec + icp$c_length ;
; 1445          1932  2
; 1446          1933  2 !
; 1447          1934  2 ! LOOP THROUGH THE LOCAL CLUSTERS AND DO THE IMAGE 0 .ADDRESS REFERENCES
; 1448          1935  2 !
; 1449          1936  2 if .adrclunum neq 0                                                      ! IF THERE ARE ANY .ADDR CLUSTER REF
; 1450          1937  3 then begin
; 1451          1938  3      dotadrvec [0] = 0 ;                                                 ! INITIALIZE THE FIXUP COUNT
; 1452          1939  3      dotadrvec [1] = 0 ;                                                 ! THE IMAGE IS NUMBER 0
; 1453          1940  3      dotadrindex   = 2 ;                                                 ! SKIP THE IMAGE NUMBER AND COUNT OF
; 1454          1941  3      incru clusterindex from 0 to .adrclunum-1                           ! LOOP THROUGH THE LIST
; 1455          1942  4      do begin
; 1456          1943  4         lnk$gl_curclu = .adrclulst [.clusterindex] ;                     ! PICK UP CURRENT CLUSTER
; 1457          1944  4         dotadrvec [0] = .dotadrvec [0] + .lnk$gl_curclu [clu$l_adrcnt] ; ! INCREMENT COUNT OF REFERENCES
; 1458          1945  4         adrblock      = lnk$gl_curclu [clu$l_fstadrl] ;                  ! GET THE LISTHEAD OF ADDRESS BLOCKS
; 1459          1946  4         while (adrblock = .adrblock [0]) neq 0                           ! LOOP THROUGH ALL THE ADDRESS BLOCK
; 1460          1947  5         do  begin
; 1461          1948  6             bind adrcnt = (if .adrblock neq .lnk$gl_curclu [clu$l_lstadrl]     ! COMPUTE COUNT OF
; 1462          1949  6                            then clu$c_adrblock - 1                             ! ELEMENTS IN THIS BLOCK
; 1463          1950  6                            else clu$c_adrblock - .lnk$gl_curclu [clu$l_adrleft] - 1
; 1464          1951  5                            ) ;
; 1465          1952  5             incru i from 1 to adrcnt                                     ! LOOP THROUGH THIS BLOCK
; 1466          1953  6             do  begin
; 1467          1954  6                 dotadrvec [.dotadrindex] = .adrblock [.i] - .lnk$gl_minva ;      ! STORE PC IN NEXT LOCATION
; 1468          1955  6                 dotadrindex = .dotadrindex + 1 ;                         ! MOVE TO NEXT LOCATION
; 1469          1956  6                 end
; 1470          1957  4             end ;
; 1471          1958  4         end
; 1472          1959  2      end ;
; 1473          1960  2
; 1474          1961  2 !
; 1475          1962  2 ! FIND ALL ISECTS THAT NEED TO BE REPROTECTED IN THE LOCAL CLUSTERS.  IF THE
; 1476          1963  2 ! CLUSTER IS A SHAREABLE IMAGE AND CONTAINS SPECIAL G^ FIXUPS, DO THOSE NOW.
; 1477          1964  2 !
; 1478          1965  2 lnk$gl_curclu = lnk$gl_clulst ;                                          ! POINT TO START OF CLUSTER LIST
; 1479          1966  2 while (lnk$gl_curclu = .lnk$gl_curclu [clu$l_nxtclu]) neq 0              ! LOOK AT ALL CLUSTERS
; 1480          1967  2 do  if not .lnk$gl_curclu [clu$v_shrimg]                                 !  BUT IGNORE SHAREABLE IMAGES
; 1481          1968  3     then begin
; 1482          1969  3          curisd = lnk$gl_curclu [clu$l_fstisd] ;                         ! POINT TO LISTHEAD
; 1483          1970  3          while (curisd = .curisd [isl$l_nxtisd]) neq 0                   ! LOOP THROUGH ALL ISECTS THIS CLUST
; 1484          1971  3          do if .curisd [isl$v_reprot]                                    ! IF THIS SECTION NEEDS TO BE REPROT
; 1485          1972  4             then begin
; 1486          1973  4                  curhdrisd = curisd [isl$t_hdrisd] ;                     ! POINT TO HEADER PART
; 1487          1974  4                  reprotvec [icp$l_baseva] = .curhdrisd [isd$v_vpn]^9     ! SET BASE VA OF ISECT
; 1488          1975  4                                             - .lnk$gl_minva ;
; 1489          1976  4                  reprotvec [icp$w_npages] = .curhdrisd [isd$w_pagcnt] ; ! AND NUMBER OF PAGES
; 1490          1977  4                  reprotvec [icp$w_newprt] = .curisd [isl$b_newprt] ;     ! COPY NEW PROTECTION
; 1491          1978  4                  reprotvec = .reprotvec + icp$c_length ;                 ! POINT TO NEXT SLOT
; 1492          1979  3                  end ;
; 1493          1980  3          end
; 1494          1981  2     else if .lnk$gl_curclu [clu$l_spcrlst] neq 0                         ! SHR IMG--IF IT CONTAINS SPECIAL G^
; 1495          1982  2          then lib$traverse_tree (lnk$gl_curclu [clu$l_spcrlst]
; 1496          1983  2                                  ,storespcgfix                           !  THEN TRAVERSE THE TREE AND STORE
; 1497          1984  2                                  ,travinfo
; 1498          1985  2                                  ) ;
; 1499       1986  2
; 1500       1987  2  !
; 1501       1988  2  ! NOW LOOP THROUGH THE CLUSTERS AND STORE THE INFORMATION
; 1502       1989  2  !
; 1503       1990  2  shrnamlist [shl$b_shl_size] = shl$c_length;                     ! Fill in length of first SHL$ entry...
; 1504       1991  2  shrnamlist                   = .shrnamlist + shl$c_length ;       ! LEAVE ROOM FOR IMAGE 0 (CURRENT IMAGE)
; 1505       1992  2  if .shrclunum neq 0
; 1506       1993  2  then incru clusterindex from 0 to .shrclunum-1                    ! FOR ALL SHAREABLE IMAGES
; 1507       1994  3      do begin
; 1508       1995  3         lnk$gl_curclu = .clusterlist [.clusterindex] ;              ! GET CLUSTER DESCRIPTOR ADDRESS
; 1509       1996  3         ch$copy (.lnk$gl_curclu [clu$b_namlng]+1                    ! COPY THE SHAREABLE IMAGE NAME
; 1510       1997  3                 ,lnk$gl_curclu [clu$b_namlng],0                     ! INTO THE SHR IMG LIST
; 1511       1998  3                 ,shl$c_maxnamlng+1
; 1512       1999  3                 ,shrnamlist [shl$t_imgnam]
; 1513       2000  3                 ) ;
; 1514       2001  3         shrnamlist  = .shrnamlist + shl$c_length ;                  ! NEXT SHR IMAGE NAME SLOT
; 1515       2002  3         shrimgindex = .shrimgindex + 1 ;                            ! NEXT IMAGE INDEX
; 1516       2003  3         if .lnk$gl_curclu [clu$l_shrsyms] neq 0                     ! IF THERE ARE G^ SYMBOLS IN
; 1517       2004  4         then begin                                                  ! THIS CLUSTER
; 1518       2005  4              local        last_symbol : initial (0) ;              ! POINTER TO LAST SYMBOL (for debbugging)
; 1519       2006  4
; 1520       2007  4              extadrvec [.extindex] = .lnk$gl_curclu [clu$l_shrsyms] ;   ! SET NUMBER OF REFERENCES
; 1521       2008  4              extindex              = .extindex + 1 ;                    ! NEXT LOCATION
; 1522       2009  4              extadrvec [.extindex] = .shrimgindex ;                     ! SET SHR IMAGE INDEX INTO G^ VECTOR
; 1523       2010  4              extindex              = .extindex + 1 ;                    ! NEXT LOCATION
; 1524       2011  4              symbolblock           = .lnk$gl_curclu [clu$l_shrlst] ;    ! GET LIST OF SHR IMG SYMBOLS THIS C
; 1525       2012  4              while .symbolblock neq 0                                   ! LOOP THROUGH THEM AND PUT THE
; 1526       2013  5              do begin                                                   ! OFFSETS INTO THE G^ VECTOR
; 1527       2014  5                 extadrvec [.extindex] = .symbolblock [sym$l_offset] ;   ! STORE OFFSET FOR THIS SYMBOL
; 1528       2015  5                 extindex              = .extindex + 1 ;                 ! NEXT G^ VECTOR LOCATION
; 1529       2016  5                 last_symbol           = .symbolblock ;                  ! REMEMBER LAST BLOCK (for debbuggin
; 1530       2017  5                 symbolblock           = .symbolblock [sym$l_shrlnk] ;   ! LINK TO NEXT SYMBOL
; 1531       2018  4                 end ;
; 1532       2019  3              end ;
; 1533       2020  3         !
; 1534       2021  3         ! IF THIS CLUSTER HAS .ADDR REFERENCES THEN STORE THEM ALSO
; 1535       2022  3         !
; 1536       2023  3         if .lnk$gl_curclu [clu$l_adrcnt] neq 0
; 1537       2024  4         then begin
; 1538       2025  4              dotadrvec [.dotadrindex]   = .lnk$gl_curclu [clu$l_adrcnt] ; ! STORE NUMBER OF SUCH REFERENCES
; 1539       2026  4              dotadrvec [.dotadrindex+1] = .shrimgindex ;                  ! SET SHR IMG INDEX INTO .ADDR VEC
; 1540       2027  4              dotadrindex                = .dotadrindex + 2 ;              ! INCREMENT THE INDEX
; 1541       2028  4              adrblock                   = lnk$gl_curclu [clu$l_fstadrl] ; ! GET FIRST .ADDR BLOCK FOR THIS C
; 1542       2029  4              while (adrblock = .adrblock [0]) neq 0                       ! LOOP THROUGH ALL .ADDR BLOCKS IN
; 1543       2030  5              do begin                                                     ! THIS CLUSTER
; 1544       2031  6                 bind adrcnt =  (if .adrblock neq .lnk$gl_curclu [clu$l_lstadrl]          ! COMPUTE NUMBER
; 1545       2032  6                                 then clu$c_adrblock - 1                                  ! OF ENTRIES IN
; 1546       2033  6                                 else clu$c_adrblock - .lnk$gl_curclu [clu$l_adrleft] - 1  ! THIS BLOCK
; 1547       2034  5                                 ) ;
; 1548       2035  5                 !
; 1549       2036  5                 ! STORE THE ADDRESSES IN THE .ADDR VECTOR
; 1550       2037  5                 !
; 1551       2038  5                 incru i from 1 to adrcnt
; 1552       2039  6                 do  begin
; 1553       2040  6                     dotadrvec [.dotadrindex] = .adrblock [.i] - .lnk$gl_minva ;
; 1554       2041  6                     dotadrindex = .dotadrindex + 1 ;
; 1555       2042  5                     end ;
; 1556          2043 4                 end ;
; 1557          2044 3              end ;
; 1558          2045 2         end ;
; 1559          2046 2
; 1560          2047 2 return true
; 1561          2048 1 end ;



                                                       0FFC 00000 LNK$CREATEFIX:
                                                                          .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11         ; 1815
                                        5E           1C  C2 00002         SUBL2   #28, SP                                      ;
                          52 00000000G  00           18  C1 00005         ADDL3   #24, LNK$GL_FIXISD, FIXISDHDR                ; 1853
                                           00000000G 00  95 0000D         TSTB    LNK$GL_CTLMSK+2                              ; 1855
                                                     04  18 00013         BGEQ    1$                                           ;
                                   0A   A2           04  88 00015         BISB2   #4, 10(FIXISDHDR)                            ; 1856
                                   09   A2           04  88 00019 1$:     BISB2   #4, 9(FIXISDHDR)                             ; 1858
                                 FF64   CF           00  FB 0001D         CALLS   #0, LNK$GETFIXSIZE                           ; 1859
                                        58           50  D0 00022         MOVL    R0, FIXSIZE                                  ;
                                   02   A2           58  B0 00025         MOVW    FIXSIZE, 2(FIXISDHDR)                        ; 1860
                          7E 00000000G  00           08  C1 00029         ADDL3   #8, LNK$GL_FIXISD, -(SP)                     ; 1861
                                                     58  DD 00031         PUSHL   FIXSIZE                                      ;
                             00000000G  00           02  FB 00033         CALLS   #2, LNK$REQUESTMEM                           ;
                                        13           50  E8 0003A         BLBS    STATUS, 2$                                   ; 1862
                                           00000000G 8F  DD 0003D         PUSHL   #LIN$_NOIMGFIL                               ; 1864
                                                     7E  D4 00043         CLRL    -(SP)                                        ;
                                                     50  DD 00045         PUSHL   STATUS                                       ;
                             00000000G  00           03  FB 00047         CALLS   #3, LIB$SIGNAL                               ;
                                                     41  11 0004E         BRB     3$                                           ; 1865
                                        50 00000000G 00  D0 00050 2$:     MOVL    LNK$GL_FIXISD, R0                            ; 1873
                          4A       14   A0           01  E0 00057         BBS     #1, 20(R0), 5$                               ;
                                        7E       02  A2  3C 0005C         MOVZWL  2(FIXISDHDR), -(SP)                          ; 1875
           50       04    A2            15           00  EF 00060         EXTZV   #0, #21, 4(FIXISDHDR), R0                    ;
                          7E            50           09  78 00066         ASHL    #9, R0, -(SP)                                ;
                             00000000G  00           02  FB 0006A         CALLS   #2, LNK$ALLOVIRMEM                           ;
                                        27           50  E8 00071         BLBS    R0, 4$                                       ;
                                           00000000G 8F  DD 00074         PUSHL   #LIN$_NOIMGFIL                               ; 1877
                          7E 00000000G  00           14  C1 0007A         ADDL3   #20, LNK$GL_IMGFIL, -(SP)                    ;
                                                     01  DD 00082         PUSHL   #1                                           ;
                                           00000000G 8F  DD 00084         PUSHL   #LIN$_CONFIXMEM                              ;
                             00000000G  00           04  FB 0008A         CALLS   #4, LIB$SIGNAL                               ;
                             00000000G  00           01  8A 00091 3$:     BICB2   #1, LNK$GL_CTLMSK                            ; 1878
                                                   02AE  31 00098         BRW     36$                                          ; 1879
                                        50 00000000G 00  D0 0009B 4$:     MOVL    LNK$GL_FIXISD, R0                            ; 1881
                                   14   A0           02  88 000A2         BISB2   #2, 20(R0)                                   ;
                                        50 00000000G 00  D0 000A6 5$:     MOVL    LNK$GL_FIXISD, R0                            ; 1884
                                        50       08  A0  D0 000AD         MOVL    8(R0), FIXBUF                                ;
                                   14   AE           50  D0 000B1         MOVL    FIXBUF, TRAVINFO                             ; 1885
           52       04    A2            15           00  EF 000B5         EXTZV   #0, #21, 4(FIXISDHDR), R2                    ; 1886
                          52            52           09  78 000BB         ASHL    #9, R2, R2                                   ;
                                   18   AE           52  D0 000BF         MOVL    R2, TRAVINFO+4                               ;
                                                     60  D4 000C3         CLRL    (FIXBUF)                                     ; 1888
                                   08   A0       40  8F  9B 000C5         MOVZBW  #64, 8(FIXBUF)                               ; 1889
           51 00000000G   00            01           02  EF 000CA         EXTZV   #2, #1, LNK$GL_CTLMSK, R1                    ; 1890
       0A  A0             01            00           51  F0 000D3         INSV    R1, #0, #1, 10(FIXBUF)                       ;
```

```
                                        55 00000000G  00  D0 000D9         MOVL     LNK$GL_SHLEXTRA, R5                         ; 1891
                                    20  A0            55  D0 000E0         MOVL     R5, 32(FIXBUF)                              ;
                     1C   A0 00000000'  EF            01  C1 000E4         ADDL3    #1, LNK$GL_SHRCLSTRS, 28(FIXBUF)            ; 1892
                                        57       40   A0  9E 000ED         MOVAB    64(R0), EXTADRVEC                           ; 1895
                                        51 00000000G  00  D0 000F1         MOVL     LNK$GL_SHRIMGS, R1                          ; 1897
                                        53 00000000G  00  D0 000F8         MOVL     LNK$GL_SHRSYMS, R3                          ;
                                        51          6341  3E 000FF         MOVAW    (R3)[R1], R1                                ;
                                        51       04 A741  DE 00103         MOVAL    4(EXTADRVEC)[R1], SPCGFIXVEC                ; 1896
                                           00000000G  00  D5 00108         TSTL     LNK$GL_SPCGFIX                              ; 1899
                                                      0F  13 0010E         BEQL     6$                                          ;
                          51 00000000G  00       18   AE  C3 00110         SUBL3    TRAVINFO+4, LNK$GL_LASTGADR, R1             ; 1900
                          54            51            50  C1 00119         ADDL3    FIXBUF, R1, REPROTVEC                       ;
                                                      03  11 0011D         BRB      7$                                          ;
                                        54            51  D0 0011F 6$:     MOVL     SPCGFIXVEC, REPROTVEC                       ; 1901
                                        56 00000000'  EF  D0 00122 7$:     MOVL     LNK$GL_RPROTISD, R6                         ; 1903
                                    08  AE       0C A446  7E 00129         MOVAQ    12(REPROTVEC)[R6], SHRNAMLIST               ;
                          51            55 00000000'  EF  C1 0012F         ADDL3    LNK$GL_SHRCLSTRS, R5, R1                    ; 1905
                          51            51            06  78 00137         ASHL     #6, R1, R1                                  ;
                                        55       08   AE  D0 0013B         MOVL     SHRNAMLIST, R5                              ; 1904
                                        6E       40 A145  9E 0013F         MOVAB    64(R1)[R5], DOTADRVEC                       ;
                     14   A0            54            50  C3 00144         SUBL3    FIXBUF, REPROTVEC, 20(FIXBUF)               ; 1910
                                                      53  D5 00149         TSTL     R3                                          ; 1912
                                                      07  13 0014B         BEQL     8$                                          ;
                     0C   A0            57            50  C3 0014D         SUBL3    FIXBUF, EXTADRVEC, 12(FIXBUF)               ; 1913
                                                      03  11 00152         BRB      9$                                          ;
                                                 0C   A0  D4 00154 8$:     CLRL     12(FIXBUF)                                  ; 1914
                                           00000000G  00  D5 00157 9$:     TSTL     LNK$GL_ADREFS                               ; 1916
                                                      07  13 0015D         BEQL     10$                                         ;
                     10   A0            6E            50  C3 0015F         SUBL3    FIXBUF, DOTADRVEC, 16(FIXBUF)               ; 1917
                                                      03  11 00164         BRB      11$                                         ;
                                                 10   A0  D4 00166 10$:    CLRL     16(FIXBUF)                                  ; 1918
                     18   A0        08  AE            50  C3 00169 11$:    SUBL3    FIXBUF, SHRNAMLIST, 24(FIXBUF)              ; 1920
                                                 04   AE  D4 0016F         CLRL     EXTINDEX                                    ; 1921
                                                 10   AE  D4 00172         CLRL     SHRIMGINDEX                                 ;
                                                      59  D4 00175         CLRL     DOTADRINDEX                                 ;
                                        84       01   A6  9E 00177         MOVAB    1(R6), (REPROTVEC)+                         ; 1926
                                        53 00000000G  00  D0 0017B         MOVL     LNK$GL_MINVA, R3                            ; 1928
                          84            52            53  C3 00182         SUBL3    R3, R2, (REPROTVEC)+                        ;
                                        84            58  B0 00186         MOVW     FIXSIZE, (REPROTVEC)+                       ; 1929
                                        84            0D  B0 00189         MOVW     #13, (REPROTVEC)+                           ; 1930
                                                 0C   AC  D5 0018C         TSTL     ADRCLUNUM                                   ; 1936
                                                      66  13 0018F         BEQL     20$                                         ;
                                                 00   BE  D4 00191         CLRL     @DOTADRVEC                                  ; 1938
                          50            6E            04  C1 00194         ADDL3    #4, DOTADRVEC, R0                           ; 1939
                                                      60  D4 00198         CLRL     (R0)                                        ;
                                        59            02  D0 0019A         MOVL     #2, DOTADRINDEX                             ; 1940
                          55        0C  AC            01  C3 0019D         SUBL3    #1, ADRCLUNUM, R5                           ; 1941
                                                      51  D4 001A2         CLRL     CLUSTERINDEX                                ; 1943
                                                      4C  11 001A4         BRB      19$                                         ;
                             00000000G  00       10 BC41  D0 001A6 12$:    MOVL     @ADRCLULST[CLUSTERINDEX], LNK$GL_CURCLU     ;
                                        58 00000000G  00  D0 001AF         MOVL     LNK$GL_CURCLU, R8                           ; 1944
                                    00  BE       34   A8  C0 001B6         ADDL2    52(R8), @DOTADRVEC                          ;
                                        56       3C   A8  9E 001BB         MOVAB    60(R8), ADRBLOCK                            ; 1945
                                        56            66  D0 001BF 13$:    MOVL     (ADRBLOCK), ADRBLOCK                        ; 1946
                                                      2C  13 001C2         BEQL     18$                                         ;
                                    40  A8            56  D1 001C4         CMPL     ADRBLOCK, 64(R8)                            ; 1948
                                                      06  13 001C8         BEQL     14$                                         ;
                                        50       7F  8F  9A  001CA        MOVZBL   #127, R0                                    ; 1949
                                                     09  11  001CE        BRB      15$                                         ;
                          50 0000007F   8F       38  A8  C3  001D0  14$:  SUBL3    56(R8), #127, R0                            ; 1950
                                        52           01  D0  001D9  15$:  MOVL     #1, I                                       ; 1952
                                                     0B  11  001DC        BRB      17$                                         ;
                    00 BE49           6642           53  C3  001DE  16$:  SUBL3    R3, (ADRBLOCK)[I], @DOTADRVEC[DOTADRINDEX]  ; 1954
                                                     59  D6  001E5        INCL     DOTADRINDEX                                 ; 1955
                                                     52  D6  001E7        INCL     I                                           ; 1952
                                        50           52  D1  001E9  17$:  CMPL     I, R0                                       ;
                                                     F0  1B  001EC        BLEQU    16$                                         ;
                                                     CF  11  001EE        BRB      13$                                         ;
                                                     51  D6  001F0  18$:  INCL     CLUSTERINDEX                                ; 1941
                                        55           51  D1  001F2  19$:  CMPL     CLUSTERINDEX, R5                            ;
                                                     AF  1B  001F5        BLEQU    12$                                         ;
                             00000000G  00 00000000G 00  9E  001F7  20$:  MOVAB    LNK$GL_CLULST, LNK$GL_CURCLU                ; 1965
                                        50 00000000G 00  D0  00202  21$:  MOVL     LNK$GL_CURCLU, R0                           ; 1966
                             00000000G  00           60  D0  00209        MOVL     (R0), [NK$GL_CURCLU                         ;
                                                     51  13  00210        BEQL     24$                                         ;
                                        50 00000000G 00  D0  00212        MOVL     LNK$GL_CURCLU, R0                           ; 1967
                          2D      58    A0           02  E0  00219        BBS      #2, 88(R0), 23$                             ;
                                        52       18  A0  9E  0021E        MOVAB    24(R0), CURISD                              ; 1969
                                        52           62  D0  00222  22$:  MOVL     (CURISD), CURISD                            ; 1970
                                                     DB  13  00225        BEQL     21$                                         ;
                                        F7       14  A2  E9  00227        BLBC     20(CURISD), 22$                             ; 1971
                                        53       18  A2  9E  0022B        MOVAB    24(R2), CURHDRISD                           ; 1973
           50       04  A3              15           00  EF  0022F        EXTZV    #0, #21, 4(CURHDRISD), R0                   ; 1974
                        50              50           09  78  00235        ASHL     #9, R0, R0                                  ;
                        84              50 00000000G 00  C3  00239        SUBL3    LNK$GL_MINVA, R0, (REPROTVEC)+              ; 1975
                                        84       02  A3  B0  00241        MOVW     2(CURHDRISD), (REPROTVEC)+                  ; 1976
                                        84       16  A2  9B  00245        MOVZBW   22(CURISD), (REPROTVEC)+                    ; 1977
                                                     D7  11  00249        BRB      22$                                         ; 1971
                                                 28  A0  D5  0024B  23$:  TSTL     40(R0)                                      ; 1981
                                                     B2  13  0024E        BEQL     21$                                         ;
                                                 14  AE  9F  00250        PUSHAB   TRAVINFO                                    ; 1982
                                               FD8B  CF  9F  00253        PUSHAB   STORESPCGFIX                                ;
                                                 28  A0  9F  00257        PUSHAB   40(R0)                                      ;
                             00000000G  00           03  FB  0025A        CALLS    #3, LIB$TRAVERSE_TREE                       ;
                                                     9F  11  00261        BRB      21$                                         ; 1967
                          50      08    AE           10  C1  00263  24$:  ADDL3    #16, SHRNAMLIST, R0                         ; 1990
                                        60       40  8F  90  00268        MOVB     #64, (R0)                                   ;
                                  08    AE 00000040  8F  C0  0026C        ADDL2    #64, SHRNAMLIST                             ; 1991
                                                 04  AC  D5  00274        TSTL     SHRCLUNUM                                   ; 1992
                                                     03  12  00277        BNEQ     25$                                         ;
                                                   00CD  31  00279        BRW      36$                                         ;
                    0C  AE        04    AC           01  C3  0027C  25$:  SUBL3    #1, SHRCLUNUM, 12(SP)                       ; 1993
                                                     5B  D4  00282        CLRL     CLUSTERINDEX                                ; 1995
                                                   00B9  31  00284        BRW      35$                                         ;
                             00000000G  00      08 BC4B  D0  00287  26$:  MOVL     @CLUSTERLIST[CLUSTERINDEX], LNK$GL_CURCLU   ;
                                        58 00000000G 00  D0  00290        MOVL     LNK$GL_CURCLU, R8                           ; 1996
                                        50       5C  A8  9A  00297        MOVZBL   92(R8), R0                                  ;
                                                     50  D6  0029B        INCL     R0                                          ;
                        7E        08    AE           18  C1  0029D        ADDL3    #24, SHRNAMLIST, -(SP)                      ; 1999
           28           00        5C    A8           50  2C  002A2        MOVC5    R0, 92(R8), #0, #40, @(SP)+                 ;
                                                     9E      002A8                                                             ;
                                  08    AE 00000040  8F  C0  002A9        ADDL2    #64, SHRNAMLIST                             ; 2001
                                                 10  AE  D6  002B1        INCL     SHRIMGINDEX                                 ; 2002
                                                 2C  A8  D5  002B4        TSTL     44(R8)                                      ; 2003
```

```
                                             35 13 002B7         BEQL   28$                                   ;
                                             50 D4 002B9         CLRL   LAST_SYMBOL                           ; 2004
                                  51       04 AE D0 002BB         MOVL   EXTINDEX, R1                          ; 2007
                                6741       2C A8 D0 002BF         MOVL   44(R8), (EXTADRVEC)[R1]               ;
                                           04 AE D6 002C4         INCL   EXTINDEX                              ; 2008
                                  51       04 AE D0 002C7         MOVL   EXTINDEX, R1                          ; 2009
                                6741       10 AE D0 002CB         MOVL   SHRIMGINDEX, (EXTADRVEC)[R1]          ;
                                           04 AE D6 002D0         INCL   EXTINDEX                              ; 2010
                                  5A       30 A8 D0 002D3         MOVL   48(R8), SYMBOLBLOCK                   ; 2011
                                             15 13 002D7 27$:    BEQL   28$                                   ; 2012
                                  51       04 AE D0 002D9         MOVL   EXTINDEX, R1                          ; 2014
                                6741       10 AA D0 002DD         MOVL   16(SYMBOLBLOCK), (EXTADRVEC)[R1]      ;
                                           04 AE D6 002E2         INCL   EXTINDEX                              ; 2015
                                  50          5A D0 002E5         MOVL   SYMBOLBLOCK, LAST_SYMBOL              ; 2016
                                  5A       1C AA D0 002E8         MOVL   28(SYMBOLBLOCK), SYMBOLBLOCK          ; 2017
                                             E9 11 002EC         BRB    27$                                   ; 2012
                                           34 A8 D5 002EE 28$:    TSTL   52(R8)                                ; 2023
                                             4B 13 002F1         BEQL   34$                                   ;
                             00 BE49       34 A8 D0 002F3         MOVL   52(R8), @DOTADRVEC[DOTADRINDEX]       ; 2025
                      50          6E          04 C1 002F9         ADDL3  #4, DOTADRVEC, R0                     ; 2026
                                6049       10 AE D0 002FD         MOVL   SHRIMGINDEX, (R0)[DOTADRINDEX]        ;
                                  59          02 C0 00302         ADDL2  #2, DOTADRINDEX                       ; 2027
                                  56       3C A8 9E 00305         MOVAB  60(R8), ADRBLOCK                      ; 2028
                                  56          66 D0 00309 29$:    MOVL   (ADRBLOCK), ADRBLOCK                  ; 2029
                                             30 13 0030C         BEQL   34$                                   ;
                              40  A8          56 D1 0030E         CMPL   ADRBLOCK, 64(R8)                      ; 2031
                                             06 13 00312         BEQL   30$                                   ;
                                  50       7F 8F 9A 00314         MOVZBL #127, R0                              ; 2032
                                             09 11 00318         BRB    31$                                   ;
                     50 0000007F  8F       38 A8 C3 0031A 30$:    SUBL3  56(R8), #127, R0                      ; 2033
                                  51          01 D0 00323 31$:    MOVL   #1, I                                 ; 2038
                                             0F 11 00326         BRB    33$                                   ;
                 00 BE49       6641 00000000G 00 C3 00328 32$:    SUBL3  LNK$GL_MINVA, (ADRBLOCK)[I], @DOTADRVEC- ; 2040
                                                                         [DOTADRINDEX]
                                             59 D6 00333         INCL   DOTADRINDEX                           ; 2041
                                             51 D6 00335         INCL   I                                     ; 2038
                                  50          51 D1 00337 33$:    CMPL   I, R0                                 ;
                                             EC 1B 0033A         BLEQU  32$                                   ;
                                             CB 11 0033C         BRB    29$                                   ; 2029
                                             5B D6 0033E 34$:    INCL   CLUSTERINDEX                          ; 1993
                              0C  AE          5B D1 00340 35$:    CMPL   CLUSTERINDEX, 12(SP)                  ;
                                             03 1A 00344         BGTRU  36$                                   ;
                                            FF3E 31 00346         BRW    26$                                   ;
                                  50          01 D0 00349 36$:    MOVL   #1, R0                                ; 2047
                                                04 0034C         RET                                          ; 2048

; Routine Size:  845 bytes,    Routine Base:  $CODE$ + 0EA6
; 1563         2049  1 routine readsection : novalue =
; 1564         2050  2 begin
; 1565         2051  2 !++
; 1566         2052  2 !                THIS ROUTINE COPIES GLOBAL SECTION FROM A SHAREABLE
; 1567         2053  2 !        IMAGE TO THE IMAGE BEING PRODUCED. IF THIS IS THE FIRST ENTRY FOR
; 1568         2054  2 !        A PARTICULAR CLUSTER THEN THE IMAGE FILE OPEN ON A PREVIUOS
; 1569         2055  2 !        CLUSTER IS CLOSED AND THIS ONE IS OPENED. THE SECTION IS READ
; 1570         2056  2 !        INTO A BUFFER WHICH IS DESCRIBED BY THE DESCRIPTOR IN THE
; 1571         2057  2 !        IMAGE SECTION DESCRIPTOR.
; 1572         2058  2 !--
; 1573         2059  2 own
; 1574         2060  2         shrimgifi,                              ! IFI OF THE SHAREABLE IMAGE FILE
; 1575         2061  2         prevauxfnb : ref block[,byte];          ! AUXILIARY FILE NAME BLOCK OF PREVIOUS FILE
; 1576         2062  2 !
; 1577         2063  2 local
; 1578         2064  2         fablock : block[fab$c_bln,byte],        ! FAB FOR OPENING AND CLOSING
; 1579         2065  2         errorcode,                              ! RMS ERROR CODE
; 1580         2066  2         stvcode,
; 1581         2067  2         bytecount;
; 1582         2068  2 !
; 1583         2069  2 bind
; 1584         2070  2         shrimgfdb = lnk$gl_curclu[clu$l_fstfdb] : ref block[,byte],
; 1585         2071  2         shrimgauxfnb = shrimgfdb[fdb$t_auxfnb] : block[,byte],
; 1586         2072  2         curhdrisd = lnk$gl_curisd[isl$t_hdrisd] : block[,byte];
; 1587         2073  2 !
; 1588         2074  2 if .shrimgifi neq 0                              ! IF A FILE IS POEN
; 1589         2075  2 and not .lnk$gl_curclu[clu$v_open]              ! AND NOT THIS FILE
; 1590         2076  3 then begin
; 1591         2077  3         $fab_init(fab=fablock);                         ! INITIALIZE THE FAB
; 1592         2078  3         fablock[fab$w_ifi] = .shrimgifi;                ! AND THE INTERNAL FILE IDENT
; 1593         2079  4         if not (errorcode = $close(fab=fablock))        ! ATTEMPT THE CLOSE
; 1594         2080  4         then begin                                      ! AND ON FAILURE
; 1595         2081  4                 signal(lin$_closein,1,shrimgfdb[fdb$g_filename],
; 1596         2082  4                                 .errorcode,.fablock[fab$l_stv]);
; 1597         2083  3                 end;
; 1598         2084  3         shrimgclu[clu$v_open] = false;                  ! FLAG FILE NOT OPEN
; 1599         2085  3         shrimgifi = 0;                                  ! RESET THE IFI
; 1600         2086  2       end;
; 1601         2087  2 if not .lnk$gl_curclu[clu$v_copy]                               ! IF THIS CLUSTER IS NOT TO BE COPIED
; 1602         2088  2 or (shrimgrab[rab$l_bkt] = .curhdrisd[isd$l_vbn]) eql 0        ! IF THIS CLUSTER NOT TO BE COPIED
; 1603         2089  3 then begin
; 1604         2090  3         lnk$gl_curisd[isl$l_bufadr] = 0;                        ! CLEAR THE BUFFER ADDRESS
; 1605         2091  3         curhdrisd[isd$l_vbn] = 0;                               ! AND THE VBN
; 1606         2092  3         return;                                         ! AND THAT'S ALL
; 1607         2093  2       end;
; 1608         2094  2 if not .lnk$gl_curclu[clu$v_open]                       ! IF THE FILE IS NOT OPEN
; 1609         2095  3 then begin
; 1610       P 2096  3         $fab_init(fab=fablock,                          ! INITIALIZE THE FAB
; 1611       P 2097  3                 fac=(bio,get),
; 1612       P 2098  3                 fop=nam,
; 1613         2099  3                 nam=shrimgauxfnb);
; 1614         2100  3         shrimgrab[rab$l_fab] = fablock;                 ! SET FAB ADDRESS IN RAB
; 1615         2101  3         shrimgrab[rab$w_isi] = 0;                       ! RESET STREAM IDENTIFIER
; 1616         2102  4         if not begin
; 1617         2103  5                 if not (errorcode = $open(fab=fablock)) ! ATTEMPT TO OPEN
; 1618         2104  5                 then begin                              ! THE IMAGE FILE AND
; 1619         2105  5                         stvcode = .fablock[fab$l_stv];  ! ON FAILURE SET STV
; 1620        2106  5                              false                                   ! AND SKIP
; 1621        2107  5                          end
; 1622        2108  4                      else true
; 1623        2109  4                      end
; 1624        2110  4              or not begin
; 1625        2111  5                      if not (errorcode = $connect(rab=shrimgrab))      ! ATTEMPT TO CONNECT RAB
; 1626        2112  5                      then begin                                        ! AND ON FAILURE
; 1627        2113  5                              stvcode = .shrimgrab[rab$l_stv];          ! SET STV CODE
; 1628        2114  5                              false                                     ! AND SKIP
; 1629        2115  5                          end
; 1630        2116  4                      else true
; 1631        2117  4                      end
; 1632        2118  4              then begin
; 1633        2119  4                      signal((lin$_openin and not sts$m_severity) or sts$k_warning,1,shrimgfdb[fdb$q_filename],
; 1634        2120  4                                      .errorcode,.stvcode);
; 1635        2121  4                      lnk$gl_curclu[clu$v_open] = false;                          ! RESET OPEN FLAG
; 1636        2122  4                      curhdrisd[isd$l_vbn] = 0;                                   ! AND THE VBN
; 1637        2123  4                      return;
; 1638        2124  3                  end;
; 1639        2125  3              shrimgifi = .fablock[fab$w_ifi];                                    ! SAVE THE IFI
; 1640        2126  3              shrimgclu = .lnk$gl_curclu;                                         ! SAVE CLUSTER DESCRIPTOR
; 1641        2127  3              shrimgclu[clu$v_open] = true;                                       ! FLAG IMAGE FILE OPEN
; 1642        2128  3              prevauxfnb = shrimgauxfnb;                                          ! THIS WILL BE PREVIOUS NEXT TIME
; 1643        2129  2          end;
; 1644        2130  2
; 1645        2131  2  shrimgrab[rab$l_ubf] = .lnk$gl_curisd[isl$l_bufadr];              ! SET BUFFER ADDRESS
; 1646        2132  2  bytecount = .curhdrisd[isd$w_pagcnt] * 512;                       ! AND BYTE COUNT
; 1647        2133  2  while .bytecount gtr 0                                  ! BEGIN LOOP
; 1648        2134  3  do begin
; 1649        2135  3          shrimgrab[rab$w_usz] = minu(.bytecount,tfrlength);        ! SET BYTE COUNT
; 1650        2136  4          if not (errorcode = $read(rab=shrimgrab))                 ! ATTEMPT NEXT READ
; 1651        2137  4          then begin
; 1652        2138  4                  signal(lin$_readerr,1,
; 1653        2139  4                          shrimgfdb[fdb$q_filename],
; 1654        2140  4                          .errorcode,.shrimgrab[rab$l_stv]);
; 1655        2141  4                  lnk$gl_curisd[isl$l_bufadr] = 0;                  ! RESET BUFFER ADDRESS
; 1656        2142  4                  curhdrisd[isd$l_vbn] = 0;                         ! AND VBN
; 1657        2143  4                  return;
; 1658        2144  3              end;
; 1659        2145  3          bytecount = .bytecount - tfrlength;                       ! REDUCE THE BYTECOUNT
; 1660        2146  3          shrimgrab[rab$l_ubf] = .shrimgrab[rab$l_ubf] +            ! ADJUST BUFFER
; 1661        2147  3                              .shrimgrab[rab$w_usz];                ! FOR AMOUNT OF READ
; 1662        2148  3          shrimgrab[rab$l_bkt] = 0;
; 1663        2149  2      end;
; 1664        2150  2  return;
; 1665        2151  1  end;            ! OF READSECTION ROUTINE


                                                                  .PSECT  $OWN$,NOEXE,2

                                                00078 SHRIMGIFI:
                                                                  .BLKB   4
                                                0007C PREVAUXFNB:
                                                                  .BLKB   4

                                                                  .EXTRN  SYS$CLOSE, SYS$OPEN
```

```
                                                               .EXTRN   SYS$READ

                                                               .PSECT   $CODE$,NOWRT,2

                                               0FFC 00000 READSECTION:
                                                               .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11    ; 2049
                               5B 00000000G  00  9E 00002      MOVAB    LNK$GL_CURCLU, R11                      ;
                               5A 00000000'  EF  9E 00009      MOVAB    SHRIMGIFI, R10                          ;
                               5E        B0  AE  9E 00010      MOVAB    -80(SP), SP                             ;
                               50            6B  D0 00014      MOVL     LNK$GL_CURCLU, R0                       ; 2070
                               58        08  A0  9E 00017      MOVAB    8(R0), R8                               ;
                  59           68            26  C1 0001B      ADDL3    #38, (R8), R9                           ; 2071
                  56 00000000' EF            18  C1 0001F      ADDL3    #24, LNK$GL_CURISD, R6                  ; 2072
                               57            6A  D0 00027      MOVL     SHRIMGIFI, R7                           ; 2074
                                             4F  13 0002A      BEQL     2$                                      ;
                  4A        58 A0            05  E0 0002C      BBS      #5, 88(R0), 2$                          ; 2075
0050   8F         00           6E            00  2C 00031      MOVC5    #0, (SP), #0, #80, $RMS_PTR             ; 2077
                                             6E     00038                                                       ;
                               6E      5003  8F  B0 00039      MOVW     #20483, $RMS_PTR                        ;
                            16 AE            02  90 0003E      MOVB     #2, $RMS_PTR+22                         ;
                            1F AE            02  90 00042      MOVB     #2, $RMS_PTR+31                         ;
                            02 AE            57  B0 00046      MOVW     R7, FABLOCK+2                           ; 2078
                                             5E  DD 0004A      PUSHL    SP                                      ; 2079
                     00000000G 00            01  FB 0004C      CALLS    #1, SYS$CLOSE                           ;
                               57            50  D0 00053      MOVL     R0, ERRORCODE                           ;
                               18            57  E8 00056      BLBS     ERRORCODE, 1$                           ;
                                         0C  AE  DD 00059      PUSHL    FABLOCK+12                              ; 2082
                                             57  DD 0005C      PUSHL    ERRORCODE                               ;
                  7E           68            14  C1 0005E      ADDL3    #20, (R8), -(SP)                        ; 2081
                                             01  DD 00062      PUSHL    #1                                      ;
                                  00000000G  8F  DD 00064      PUSHL    #LIN$_CLOSEIN                           ;
                     00000000G 00            05  FB 0006A      CALLS    #5, LIB$SIGNAL                          ;
                               50        D8  AA  D0 00071 1$:  MOVL     SHRIMGCLU, R0                           ; 2084
                            58 A0            20  8A 00075      BICB2    #32, 88(R0)                             ;
                                             6A  D4 00079      CLRL     SHRIMGIFI                               ; 2085
                               51            6B  D0 0007B 2$:  MOVL     LNK$GL_CURCLU, R1                       ; 2087
                  03        58 A1            04  E0 0007E      BBS      #4, 88(R1), 4$                          ;
                                           00E9  31 00083 3$:  BRW      12$                                     ;
                            C8 AA        0C  A6  D0 00086 4$:  MOVL     12(R6), SHRIMGRAB+56                    ; 2088
                                             F6  13 0008B      BEQL     3$                                      ;
                  03        58 A1            05  E1 0008D      BBC      #5, 88(R1), 5$                          ; 2094
                                           0085  31 00092      BRW      9$                                      ;
0050   8F         00           6E            00  2C 00095 5$:  MOVC5    #0, (SP), #0, #80, $RMS_PTR             ; 2099
                                             6E     0009C                                                       ;
                               6E      5003  8F  B0 0009D      MOVW     #20483, $RMS_PTR                        ;
                            04 AE  01000000  8F  D0 000A2      MOVL     #16777216, $RMS_PTR+4                   ;
                            16 AE            22  90 000AA      MOVB     #34, $RMS_PTR+22                        ;
                            1F AE            02  90 000AE      MOVB     #2, $RMS_PTR+31                         ;
                            28 AE            59  D0 000B2      MOVL     R9, $RMS_PTR+40                         ;
                            CC AA            6E  9E 000B6      MOVAB    FABLOCK, SHRIMGRAB+60                   ; 2100
                                         92  AA  B4 000BA      CLRW     SHRIMGRAB+2                             ; 2101
                                             5E  DD 000BD      PUSHL    SP                                      ; 2103
                     00000000G 00            01  FB 000BF      CALLS    #1, SYS$OPEN                            ;
                               57            50  D0 000C6      MOVL     R0, ERRORCODE                           ;
                               06            57  E8 000C9      BLBS     ERRORCODE, 6$                           ;
                               52        0C  AE  D0 000CC      MOVL     FABLOCK+12, STVCODE                     ; 2105
                                             14  11 000D0      BRB      7$                                      ;
```

```
                       90 AA 9F 000D2 6$:   PUSHAB   SHRIMGRAB                                      ; 2111
   00000000G 00           01 FB 000D5       CALLS    #1, SYS$CONNECT                                ;
             57           50 D0 000DC       MOVL     R0, ERRORCODE                                  ;
             24           57 E8 000DF       BLBS     ERRORCODE, 8$                                  ;
             52        9C AA D0 000E2       MOVL     SHRIMGRAB+12, STVCODE                          ; 2113
                          52 DD 000E6 7$:   PUSHL    STVCODE                                        ; 2120
                          57 DD 000E8       PUSHL    ERRORCODE                                      ;
7E           68           14 C1 000EA       ADDL3    #20, (R8), -(SP)                               ; 2119
                          01 DD 000EE       PUSHL    #1                                             ;
                00000000* 8F DD 000F0       PUSHL    #<LIN$_OPENIN&-8>                              ;
   00000000G 00           05 FB 000F6       CALLS    #5, LIB$SIGNAL                                 ;
             50           6B D0 000FD       MOVL     LNK$GL_CURCLU, R0                              ; 2121
          58 A0           20 8A 00100       BICB2    #32, 88(R0)                                    ;
                          73 11 00104       BRB      13$                                            ; 2122
             6A        02 AE 3C 00106 8$:   MOVZWL   FABLOCK+2, SHRIMGIFI                           ; 2125
          D8 AA           6B D0 0010A       MOVL     LNK$GL_CURCLU, SHRIMGCLU                       ; 2126
             50        D8 AA D0 0010E       MOVL     SHRIMGCLU, R0                                  ; 2127
          58 A0           20 88 00112       BISB2    #32, 88(R0)                                    ;
          04 AA           59 D0 00116       MOVL     R9, PREVAUXFNB                                 ; 2128
             50 00000000' EF D0 0011A 9$:   MOVL     LNK$GL_CURISD, R0                              ; 2131
          B4 AA        08 A0 D0 00121       MOVL     8(R0), SHRIMGRAB+36                            ;
             52        02 A6 3C 00126       MOVZWL   2(R6), BYTECOUNT                               ; 2132
52           52           09 78 0012A       ASHL     #9, BYTECOUNT, BYTECOUNT                       ;
                          52 D5 0012E 10$:  TSTL     BYTECOUNT                                      ; 2133
                          5D 15 00130       BLEQ     15$                                            ;
             50           52 D0 00132       MOVL     BYTECOUNT, R0                                  ; 2135
    00001400 8F           50 D1 00135       CMPL     R0, #5120                                      ;
                          05 1B 0013C       BLEQU    11$                                            ;
             50      1400 8F 3C 0013E       MOVZWL   #5120, R0                                      ;
          B0 AA           50 B0 00143 11$:  MOVW     R0, SHRIMGRAB+32                               ;
                       90 AA 9F 00147       PUSHAB   SHRIMGRAB                                      ; 2136
   00000000G 00           01 FB 0014A       CALLS    #1, SYS$READ                                   ;
             57           50 D0 00151       MOVL     R0, ERRORCODE                                  ;
             26           57 E8 00154       BLBS     ERRORCODE, 14$                                 ;
                       9C AA DD 00157       PUSHL    SHRIMGRAB+12                                   ; 2140
                          57 DD 0015A       PUSHL    ERRORCODE                                      ;
7E           68           14 C1 0015C       ADDL3    #20, (R8), -(SP)                               ; 2139
                          01 DD 00160       PUSHL    #1                                             ;
                00000000G 8F DD 00162       PUSHL    #LIN$_READERR                                  ;
   00000000G 00           05 FB 00168       CALLS    #5, LIB$SIGNAL                                 ;
             50 00000000' EF D0 0016F 12$:  MOVL     LNK$GL_CURISD, R0                              ; 2141
                       08 A0 D4 00176       CLRL     8(R0)                                          ;
                       0C A6 D4 00179 13$:  CLRL     12(R6)                                         ; 2142
                             04 0017C       RET                                                     ; 2137
             52      EC00 C2 9E 0017D 14$:  MOVAB    -5120(R2), BYTECOUNT                           ; 2145
             50        B0 AA 3C 00182       MOVZWL   SHRIMGRAB+32, R0                               ; 2147
          B4 AA           50 C0 00186       ADDL2    R0, SHRIMGRAB+36                               ;
                       C8 AA D4 0018A       CLRL     SHRIMGRAB+56                                   ; 2148
                          9F 11 0018D       BRB      10$                                            ; 2133
                             04 0018F 15$:  RET                                                     ; 2151

; Routine Size:  400 bytes,    Routine Base:  $CODE$ + 11F3
; 1667          2152  1 routine writesection(pages,vbn,buffer) =
; 1668          2153  2 begin
; 1669          2154  2 local
; 1670          2155  2         bytecount,                                                          ! NUMBER OF BYTES OF IMAGE FILE
; 1671          2156  2         errorcode;                                                          ! WRITE ERROR CODE RETURNED
; 1672          2157  2 !
; 1673          2158  2 bind auxfnb = lnk$gl_imgfil[fdb$t_auxfnb] : block[nam$c_bln,byte];         ! AUXILIARY FILENAME BLOCK
; 1674          2159  2 !
; 1675          2160  2 bytecount = .pages * 512;
; 1676          2161  2 lnk$al_imgrab[rab$l_bkt] = .vbn;
; 1677          2162  2 lnk$al_imgrab[rab$l_rbf] = .buffer;
; 1678          2163  2 while .bytecount gtr 0                                                      ! LOOP TILL ALL DONE
; 1679          2164  3 do begin
; 1680          2165  3         lnk$al_imgrab[rab$w_rsz] = min(.bytecount,tfrlength);               ! SET BYTE COUNT FOR THIS WRITE
; 1681          2166  4         if not (errorcode = $write(rab=lnk$al_imgrab))                      ! AND DO IT
; 1682          2167  4         then begin                                                          ! BUT IF AN
; 1683          2168  4                 signal(lin$_writeerr,1,lnk$gl_imgfil[fdb$q_filename],       ! AND REPORT THE ERROR
; 1684          2169  4                         .errorcode,.lnk$al_imgrab[rab$l_stv],lin$_noimgfil);
; 1685          2170  4                 lnk$gl_ctlmsk[lnk$v_image] = false;                         ! MESSAGE
; 1686          2171  4                 return false;                                               ! THEN GIVE UP
; 1687          2172  3             end;
; 1688          2173  3         bytecount = .bytecount - tfrlength;                                 ! REDUCE THE BYTE COUNT
; 1689          2174  3         lnk$al_imgrab[rab$l_rbf] = .lnk$al_imgrab[rab$l_rbf] +              ! UPDATE BUFFER POINTER
; 1690          2175  3                                 .lnk$al_imgrab[rab$w_rsz];                  ! TO THE NEXT PIECE
; 1691          2176  3         lnk$al_imgrab[rab$l_bkt] = 0;                                       ! FORCE RMS TO USE NEXT BLOCK POINTE
; 1692          2177  2     end;                                                                    ! AND GO AGAIN
; 1693          2178  2 return true;
; 1694          2179  1 end;



                                                       .EXTRN  SYS$WRITE

                                001C 00000 WRITESECTION:
                                                       .WORD   Save R2,R3,R4                            ; 2152
             54 00000000'  EF   9E 00002               MOVAB   LNK$AL_IMGRAB+56, R4                     :
52       04 AC             09   78 00009               ASHL    #9, PAGES, BYTECOUNT                     ; 2160
            64       08    AC   D0 0000E               MOVL    VBN, LNK$AL_IMGRAB+56                    ; 2161
     F0     A4       0C    AC   D0 00012               MOVL    BUFFER, LNK$AL_IMGRAB+40                 ; 2162
                           52   D5 00017 1$:           TSTL    BYTECOUNT                                ; 2163
                           61   15 00019               BLEQ    4$                                       :
            50             52   D0 0001B               MOVL    BYTECOUNT, R0                            ; 2165
   00001400 8F             50   D1 0001E               CMPL    R0, #5120                                :
                           05   15 00025               BLEQ    2$                                       :
            50     1400    8F   3C 00027               MOVZWL  #5120, R0                                :
     EA     A4             50   B0 0002C 2$:           MOVW    R0, LNK$AL_IMGRAB+34                     :
                     C8    A4   9F 00030               PUSHAB  LNK$AL_IMGRAB                            ; 2166
   00000000G 00            01   FB 00033               CALLS   #1, SYS$WRITE                            :
            53             50   D0 0003A               MOVL    R0, ERRORCODE                            :
            2B             53   E8 0003D               BLBS    ERRORCODE, 3$                            :
              00000000G    8F   DD 00040               PUSHL   #LIN$_NOIMGFIL                           ; 2168
                     D4    A4   DD 00046               PUSHL   LNK$AL_IMGRAB+12                         ; 2169
                           53   DD 00049               PUSHL   ERRORCODE                                :
7E 00000000G 00            14   C1 0004B               ADDL3   #20, LNK$GL_IMGFIL, -(SP)                ; 2168
                           01   DD 00053               PUSHL   #1                                       :
              00000000G    8F   DD 00055               PUSHL   #LIN$_WRITEERR                           :
   00000000G 00            06   FB 0005B               CALLS   #6, LIB$SIGNAL                           :
          00000000G  00           01  8A 00062         BICB2   #1, LNK$GL_CTLMSK              ; 2170
                                  15  11 00069         BRB     5$                             ; 2171
                     52     EC00  C2  9E 0006B 3$:     MOVAB   -5120(R2), BYTECOUNT           ; 2173
                     50       EA  A4  3C 00070         MOVZWL  LNK$AL_IMGRAB+34, R0           ; 2175
              F0     A4           50  C0 00074         ADDL2   R0, LNK$AL_IMGRAB+40           ;
                                  64  D4 00078         CLRL    LNK$AL_IMGRAB+56               ; 2176
                                  9B  11 0007A         BRB     1$                             ; 2163
                     50           01  D0 0007C 4$:     MOVL    #1, R0                         ; 2178
                                      04 0007F         RET                                    ;
                                  50  D4 00080 5$:     CLRL    R0                             ; 2179
                                      04 00082         RET                                    ;

; Routine Size:  131 bytes,    Routine Base:  $CODE$ + 1383
; 1696         2180 1 global routine lnk$wrtimghdr : novalue =
; 1697         2181 2 begin
; 1698         2182 2 !
; 1699         2183 2 ! THIS ROUTINE IS CALLED TO WRITE THE IMAGE HEADER. IT FIRST WRITES
; 1700         2184 2 ! THE HEADER FIXED RECORD INTO THE FIRST HEADER BUFFER BLOCK THEN
; 1701         2185 2 ! OUTPUTS TO THE (OPEN) IMAGE FILE ALL BLOCKS OF THE IMAGE HEADER
; 1702         2186 2 ! FINALLY THE IMAGE FILE IS CLOSED.
; 1703         2187 2 !
; 1704         2188 2 global
; 1705         2189 2     lnk$gl_otsepdsc      : ref block [,byte] ;                   ! POINTER TO DESCRIPTOR OF OTS ENTRY POINT
; 1706         2190 2 !
; 1707         2191 2 local
; 1708         2192 2     curtfr,                                                     ! ADDRESS IN HEADER FOR NEXT TRANSFER ADDRES
; 1709         2193 2     otsepnam             : ref block [,byte],                    ! POINTER TO OTS EPT NAME BLOCK
; 1710         2194 2     sysver               : ref block [,byte],                    ! POINTER TO VALUE PART OF SYS$K_VERSION
; 1711         2195 2     sysvernam            : ref block [,byte],                    ! POINTER TO NAME PART OF SYS$K_VERSION
; 1712         2196 2     ihdactiv             : ref block [,byte] ;                   ! IMAGE ACTIVATION AREA OF HEADER
; 1713         2197 2 !
; 1714         2198 2 bind
; 1715         2199 2     otsentrysym = cstring ('LIB$INITIALIZE'),                    ! OTS ENTRY POINT NAME
; 1716         2200 2     otsinishr   = cstring ('LIB$IMAGE_INISHR'),                  ! OTS ENTRY POINT FOR SHARED IMAGES
; 1717         2201 2     imgidarea   = ihdactiv                : ref block [,byte],  ! USE SAME CELL TO POINT TO ID AREA
; 1718         2202 2     symdbgdata  = ihdactiv                : ref block [,byte],  ! AND FOR SYMBOL TABLE AND DEBUG DATA
; 1719         2203 2     imageheader = header_desc [0]         : ref block [,byte],  ! POINT TO THE BUFFER
; 1720         2204 2     imageprivs  = imageheader [ihd$q_privreqs]  : vector [2,long] ;      ! POINT TO PRIVILEGES
; 1721         2205 2 !
; 1722         2206 3 if (.lnk$gl_ctlmsk [lnk$v_sys]                                   ! IF A SYSTEM IMAGE
; 1723         2207 3         and not
; 1724         2208 3     .lnk$gl_ctlmsk [lnk$v_sysheadr]                             !  WITHOUT A HEADER
; 1725         2209 3    )
; 1726         2210 2 or  .lnk$gw_hdrblks eql 0                                        ! OR THERE IS NO HEADER
; 1727         2211 3 then begin
; 1728         2212 3     lnk$closimgfil () ;                                         ! CLOSE THE IMAGE FILE
; 1729         2213 3     return ;                                                    ! AND THAT'S ALL
; 1730         2214 3     end ;
; 1731         2215 2
; 1732         2216 2 imageheader [ihd$b_hdrblkcnt] = .lnk$gw_hdrblks ;                ! INSERT NUMBER OF BLOCKS OF HEADER IN HEADE
; 1733         2217 2 imageheader [ihd$w_size]      =  ihd$k_maxlength ;               ! SIZE OF HEADER
; 1734         2218 2 imageheader [ihd$w_activoff]  =  ihd$k_activoff ;                ! OFFSET TO ACTIVATION CONTROL
; 1735         2219 2 imageheader [ihd$w_symdbgoff] =  ihd$k_symdbgoff ;               ! SYMBOL AND DEBUG DATA
; 1736         2220 2 imageheader [ihd$w_imgidoff]  =  ihd$k_imgidoff ;                ! THE IDENTIFICATION AREA
; 1737         2221 2 imageheader [ihd$b_imgtype]   =  ihd$k_exe ;                     ! ASSUME EXECUTABLE IMAGE
; 1738         2222 2 imageheader [ihd$w_iochancnt] = .lnk$gw_chans ;                  ! SET NUMBER OF CHANNELS
; 1739         2223 2 imageheader [ihd$w_imgiocnt]  = .lnk$gw_ioseg ;                  ! AND SIZE OF I/O SEGMENT
; 1740         2224 2 imageheader [ihd$v_lnkdebug]  = .lnk$gl_ctlmsk [lnk$v_dbg] ;     ! AND THE STATE OF DEBUG FLAG
; 1741         2225 2 imageheader [ihd$v_picimg]    = .lnk$gl_ctlmsk [lnk$v_picimg] ;  ! EXTRACT POSITION DEPENDENCE FLAG
; 1742         2226 2 imageheader [ihd$v_nop0bufs]  = .lnk$gl_ctlmsk [lnk$v_nop0bufs] ;  ! AND STATE OF RMS P0 BUFFER INDICATOR
; 1743         2227 2 imageheader [ihd$v_matchctl]  = .lnk$gb_matchctl ;               ! MATCH CONTROL FLAGS
; 1744         2228 2 imageheader [ihd$v_dbgdmt]    =  true ;                          ! DMT FIELDS ARE PRESENT
; 1745         2229 2 imageheader [ihd$l_ident]     = .lnk$gl_matchid ;                ! AND THE ASSOCIATED IDENT
; 1746         2230 2
; 1747         2231 2 imageprivs [0] = .lnk$gq_privs [0] ;                             ! SET PRIVS INTO HEADER
; 1748         2232 2 imageprivs [1] = .lnk$gq_privs [1] ;                             ! ...
; 1749         2233 2
; 1750         2234 2
; 1751         2235 3 imageheader [ihd$l_sysver] = (if lnk$search (lnk$gt_sysver,sysver,sysvernam)    ! LOOK UP SYS$K_VERSION
; 1752         2236 3                               then  .sysver [sym$l_value]      ! AND STORE THE VALUE
; 1753      2237  3                                 else 0                            ! WE REALLY SHOULD ALWAYS FIND IT!
; 1754      2238  2                                ) ;
; 1755      2239  3  imageheader [ihd$l_iafva]  = (if .lnk$gl_fixisd neq 0            ! IF THERE IS A FIXUP SECTION
; 1756      2240  4                                 then begin
; 1757      2241  4                                      bind fixhdrisd = lnk$gl_fixisd [isl$t_hdrisd] : block [,byte] ;
; 1758      2242  4                                      .fixhdrisd [isd$v_vpn]^9             ! SET VA OF FIXUP SECTION
; 1759      2243  4                                      end
; 1760      2244  3                                 else 0                                    ! ELSE USE 0
; 1761      2245  2                                ) ;
; 1762      2246  2  imageheader [ihd$v_p0image] = .lnk$gl_ctlmsk [lnk$v_p0image] ;   ! SET PO-ONLY STATUS
; 1763      2247  2  ihdactiv = .imageheader + .imageheader [ihd$w_activoff] ;       ! POINT TO ACTIVATION AREA
; 1764      2248  2  curtfr   = ihdactiv [iha$l_tfradr1] ;                           ! START AT FIRST TRANSFER ADDRESS
; 1765      2249  2
; 1766      2250  2  if .lnk$gl_dbgtfps neq 0                                         !  IF A
; 1767      2251  3  then begin                                                       ! DEBUGGER TRANSFER
; 1768      2252  3      .curtfr = .lnk$gl_dbgtfr ;                                   ! ADDRESS WAS FOUND
; 1769      2253  3       curtfr = ihdactiv [iha$l_tfradr2] ;                         ! THEN IT IS FIRST AND UPDATE TO NEXT
; 1770      2254  3       end
; 1771      2255  2  else if .lnk$gl_ctlmsk [lnk$v_dbg]                               ! NO DEBUGGER TRANSFER ADDRESS SO
; 1772      2256  2       then  signal (lin$_dbgtfr,1                                 ! CHECK IF DEBUG WAS REQUESTED
; 1773      2257  2                   ,lnk$gl_imgfil [fdb$q_filename]                 ! AND ISSUE ERROR IF SO
; 1774      2258  2                    ) ;
; 1775      2259  2
; 1776      2260  2  ihdactiv [iha$l_inishr] = 0 ;                                    ! ASSUME NO SHAREABLE IMAGE INITIALIZATION
; 1777      2261  2  if .lnk$gl_ctlmsk [lnk$v_shr]                                    ! IF SHAREABLE IMAGE
; 1778      2262  3  then begin
; 1779      2263  3       imageheader [ihd$b_imgtype] = ihd$k_shr ;                   ! SET TO SHAREABLE
; 1780      2264  3       if lnk$search (otsinishr,lnk$gl_otsepdsc,otsepnam)          ! IF ''LIB$IMAGE_INISHR'' SPECIFIED
; 1781      2265  4       then begin
; 1782      2266  4           ihdactiv [iha$l_inishr] = .lnk$gl_otsepdsc [sym$l_value]  ; ! SETUP INIT ENTRY POINT
; 1783      2267  4           imageheader [ihd$v_inishr] = true ;                         ! MARK IT VALID
; 1784      2268  3            end ;
; 1785      2269  2       end ;
; 1786      2270  2
; 1787      2271  2
; 1788      2272  2  if not .lnk$gl_ctlmsk [lnk$v_sys]                                ! FOR A NON SYSTEM IMAGE
; 1789      2273  2  then if lnk$search (otsentrysym,lnk$gl_otsepdsc,otsepnam)        ! IF THE RESERVED EPT NAME FOR
; 1790      2274  3       then begin                                                  ! THE OTS HAS BEEN DEFINED
; 1791      2275  3            .curtfr = .lnk$gl_otsepdsc [sym$l_value] ;             ! THEN NEXT TRANSFER IS TO THE OTS
; 1792      2276  3           if  .curtfr eql ihdactiv [iha$l_tfradr2]                ! AND UPDATE THE POINTER FOR THE
; 1793      2277  3           then curtfr = ihdactiv [iha$l_tfradr3]                  ! USER TRANSFER ADDRESS TO BE THE SECOND
; 1794      2278  3           else curtfr = ihdactiv [iha$l_tfradr2] ;                ! OR THIRD DEPENDING ON PRESENCE OF DEBUGGER
; 1795      2279  2           end ;                                                   ! BEFORE THE OTS
; 1796      2280  2
; 1797      2281  2  imageheader [ihd$v_lnknotfr] = true ;                            ! ASSUME NO TRANSFER ADDRESS
; 1798      2282  2
; 1799      2283  2  if .lnk$gl_tfrpsc neq 0                                          ! FINALLY IF THE USER HAD A
; 1800      2284  3  then begin                                                       ! A TRANSFER ADDRESS
; 1801      2285  3       imageheader [ihd$v_lnknotfr] = false ;                      ! CLEAR NO TRANSFER ADDRESS FLAG
; 1802      2286  4       if not  (.lnk$gl_ctlmsk [lnk$v_sys]                         ! IF NOT A SYSTEM IMAGE
; 1803      2287  4                and not
; 1804      2288  4               .lnk$gl_ctlmsk [lnk$v_sysheadr]
; 1805      2289  4              )                                                    !    WITH NO HEADER
; 1806      2290  3       then .curtfr = .lnk$gl_tfradr                               ! AND IF SO INSERT IT LAST
; 1807      2291  3       else  signal (lin$_tfrsys,1                                 ! REPORT IGNORED TRANSFER ADDRESS
; 1808      2292  3                   ,lnk$gl_imgfil [fdb$q_filename]                 ! IN OTHER KINDS OF IMAGE
; 1809      2293  3                    ) ;
; 1810          2294  3      end
; 1811          2295  2 else if .lnk$gl_ctlmsk [lnk$v_exe]                             ! IF IT IS AN EXECUTABLE
; 1812          2296  2      then signal (lin$_usrtfr,1                                ! IMAGE REPORT THE ERROR
; 1813          2297  2                   ,lnk$gl_imgfil [fdb$q_filename]              ! WHEN TRANSFER ADDRESS IS MISSING
; 1814          2298  2                   ) ;
; 1815          2299  2
; 1816          2300  2 symdbgdata = .imageheader + .imageheader [ihd$w_symdbgoff] ;     ! POINT TO STB AND DBG DATA AREA
; 1817          2301  2
; 1818          2302  2 if  (symdbgdata [ihs$w_dstblks] = .lnk$gw_dstblks) neq 0         ! IF A DST WAS WRITTEN
; 1819          2303  2 then symdbgdata [ihs$l_dstvbn]  = .lnk$gw_dstvbn ;               ! SET BLOCK COUNT AND ITS VBN
; 1820          2304  2
; 1821          2305  2 if  (symdbgdata [ihs$w_gstrecs] = .lnk$gw_gstrecs) neq 0         ! IF A GST WAS WRITTEN
; 1822          2306  2 then symdbgdata [ihs$l_gstvbn]  = .lnk$gw_gstvbn ;               ! THEN SET RECORD COUNT AND VBN IN HEADER
; 1823          2307  2
; 1824          2308  2 if  (symdbgdata [ihs$l_dmtbytes] = .lnk$gl_dmtbytes) neq 0
; 1825          2309  2 then symdbgdata [ihs$l_dmtvbn]   = .dmtvbn ;
; 1826          2310  2
; 1827          2311  2 !
; 1828          2312  2 ! THE MAJOR AND MINOR IDENTS OF THE HEADER RECORD ARE USED
; 1829          2313  2 ! BY IMAGE ACTIVATOR AS FOLLOWS:-
; 1830          2314  2 !                       MAJOR ID - MATCH REQUIRED ALWAYS
; 1831          2315  2 !                       MINOR ID - MORE RECENT (GREATER THAN
; 1832          2316  2 !                       OR EQUAL) THAN IMAGE ACTIVATOR'S
; 1833          2317  2 !                       VERSION ARE PERMITTED.
; 1834          2318  2 !
; 1835          2319  2 imageheader [ihd$w_majorid] = ihd$k_majorid ;                                     ! MAJOR IDENT
; 1836          2320  2 imageheader [ihd$w_minorid] = ihd$k_minorid ;                                     ! MINOR IDENT
; 1837          2321  2
; 1838          2322  2 imgidarea = .imageheader + .imageheader [ihd$w_imgidoff] ;                        ! POINT TO IMAGE ID AREA
; 1839          2323  2
; 1840          2324  2 lnk$gt_imgnam [0] = minu (lnk$k_max_filename_length,.lnk$gt_imgnam [0]) ;        ! TRIM NAME LENGTH TO MAX
; 1841          2325  2 lnk$gt_imgid [0]  = minu (sym$c_shortname,.lnk$gt_imgid [0]) ;                   ! ALLOWED IN HEADER
; 1842          2326  2
; 1843          2327  2 ch$move (.lnk$gt_imgnam [0]+1,lnk$gt_imgnam [0],imgidarea [ihi$t_imgnam]) ;     ! MOVE IN THE IMAGE NAME
; 1844          2328  2 ch$move (.lnk$gt_imgid [0]+1, lnk$gt_imgid [0], imgidarea [ihi$t_imgid]) ;     ! MOVE IN THE IMAGE I.D
; 1845          2329  2
; 1846          2330  2 (imgidarea [ihi$q_linktime])<0,32,0> = .lnk$gq_alostim [0] ;                   ! TIME OF LINK IS WHEN
; 1847          2331  2 imgidarea [ihi$q_linktime] + 4      = .lnk$gq_alostim [1] ;                   ! MEMORY ALLOCATION BEGAN
; 1848          2332  2 (imgidarea [ihi$t_linkid])<0,8,0>   =  lid$c_size + 1 ;                       ! INSERT LENGTH OF LINKER ID
; 1849          2333  2 imgidarea [ihi$t_linkid] + 1        = .lnk$aw_version [lid$w_major] ;         ! FOLLOWED BY THE MAJOR ID
; 1850          2334  2 imgidarea [ihi$t_linkid] + 3        = %c'-' ;                                 ! SEPARETED BY A HYPHEN
; 1851          2335  2 imgidarea [ihi$t_linkid] + 4        = .lnk$aw_version [lid$w_minor] ;         ! FROM THE MINOR ID
; 1852          2336  2
; 1853          2337  2 !
; 1854          2338  2 ! FINALLY WRITE OUT THE WHOLE HEADER.
; 1855          2339  2 !
; 1856          2340  2 if writesection (.lnk$gw_hdrblks, 1, .header_desc [0])           ! IF SUCCESSFULLY WRITTEN TO VBN 1
; 1857          2341  2 then lnk$closimgfil () ;                                           ! CLOSE THE FILE
; 1858          2342  2 return ;
; 1859          2343  1 end ;


                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                     0E  0004B P.AAG:   .BYTE   14
   45  5A  49  4C  41  49  54  49  4E  49  24  42  49  4C  0004C            .ASCII  \LIB$INITIALIZE\                      ;
```

```
                                                         10 0005A P.AAH:  .BYTE   16                                          ;
48  53  49  4E  49  5F  45  47  41  4D  49  24  42  49   4C 0005B         .ASCII  \LIB$IMAGE_INISHR\                          ;
                                                         52 0006A                                                             ;

                                                                          .PSECT  $GLOBAL$,NOEXE,2

                                                            00079         .BLKB   3
                                                            0007C LNK$GL_OTSEPDSC::
                                                                          .BLKB   4

                                                                  OTSENTRYSYM=        P.AAG
                                                                  OTSINISHR=          P.AAH
                                                                  IMAGEHEADER=        HEADER_DESC

                                                                          .PSECT  $CODE$,NOWRT,2

                                                       0FFC 00000         .ENTRY  LNK$WRTIMGHDR, Save R2,R3,R4,R5,R6,R7,R8,-  ; 2180
                                                                                  R9,R10,R11                                  ;
                                       5B 00000000G  00  9E 00002         MOVAB   LNK$GT_IMGNAM, R11                          ;
                                       5A 00000000'  EF  9E 00009         MOVAB   IMAGEHEADER, R10                            ;
                                       59 00000000'  EF  9E 00010         MOVAB   LNK$GL_OTSEPDSC, R9                         ;
                                       58 00000000G  00  9E 00017         MOVAB   LNK$GL_CTLMSK, R8                           ;
                                       5E            0C  C2 0001E         SUBL2   #12, SP                                     ;
                         53            6A            14  C1 00021         ADDL3   #20, IMAGEHEADER, R3                        ; 2204
                         08            68            03  E1 00025         BBC     #3, LNK$GL_CTLMSK, 2$                       ; 2206
                         03       03   A8            01  E0 00029         BBS     #1, LNK$GL_CTLMSK+3, 2$                     ; 2208
                                                   0257  31 0002E 1$:     BRW     22$                                         ;
                                                EC   A9  B5 00031 2$:     TSTW    LNK$GW_HDRBLKS                              ; 2210
                                                     F8  13 00034         BEQL    1$                                          ;
                                       52            6A  D0 00036         MOVL    IMAGEHEADER, R2                             ; 2216
                                  10   A2       EC   A9  90 00039         MOVB    LNK$GW_HDRBLKS, 16(R2)                      ;
                                       62  003000A8  8F  D0 0003E         MOVL    #3145896, (R2)                              ; 2217
                                  04   A2  00580044  8F  D0 00045         MOVL    #5767236, 4(R2)                             ; 2219
                                  11   A2            01  90 0004D         MOVB    #1, 17(R2)                                  ; 2221
                                  1C   A2 00000000G  00  B0 00051         MOVW    LNK$GW_CHANS, 28(R2)                        ; 2222
                                  1E   A2 00000000G  00  B0 00059         MOVW    LNK$GW_IOSEG, 30(R2)                        ; 2223
                                  50            20   A2  9E 00061         MOVAB   32(R2), R0                                  ; 2224
          51             68            01            06  EF 00065         EXTZV   #6, #1, LNK$GL_CTLMSK, R1                   ;
          60             01            00            51  F0 0006A         INSV    R1, #0, #1, (R0)                            ;
          51        02   A8            01            01  EF 0006F         EXTZV   #1, #1, LNK$GL_CTLMSK+2, R1                 ; 2225
          60             01            03            51  F0 00075         INSV    R1, #3, #1, (R0)                            ;
          51        02   A8            01            05  EF 0007A         EXTZV   #5, #1, LNK$GL_CTLMSK+2, R1                 ; 2226
          60             01            02            51  F0 00080         INSV    R1, #2, #1, (R0)                            ;
      03  A0             03            00       FC   A9  F0 00085         INSV    LNK$GB_MATCHCTL, #0, #3, 3(R0)              ; 2227
                                       60            20  88 0008C         BISB2   #32, (R0)                                   ; 2228
                                  24   A2       F8   A9  D0 0008F         MOVL    LNK$GL_MATCHID, 36(R2)                      ; 2229
                                       63 00000000G  00  7D 00094         MOVQ    LNK$GQ_PRIVS, (R3)                          ; 2231
                                                     5E  DD 0009B         PUSHL   SP                                          ; 2235
                                                08   AE  9F 0009D         PUSHAB  SYSVER                                      ;
                                          00000000G  00  9F 000A0         PUSHAB  LNK$GT_SYSVER                               ;
                            00000000G  00            03  FB 000A6         CALLS   #3, LNK$SEARCH                              ;
                                       06            50  E9 000AD         BLBC    R0, 3$                                      ;
                                       50       04   BE  D0 000B0         MOVL    @SYSVER, R0                                 ; 2236
                                                     02  11 000B4         BRB     4$                                          ;
                                                     50  D4 000B6 3$:     CLRL    R0                                          ; 2235
                                  28   A2            50  D0 000B8 4$:     MOVL    R0, 40(R2)                                  ;
```

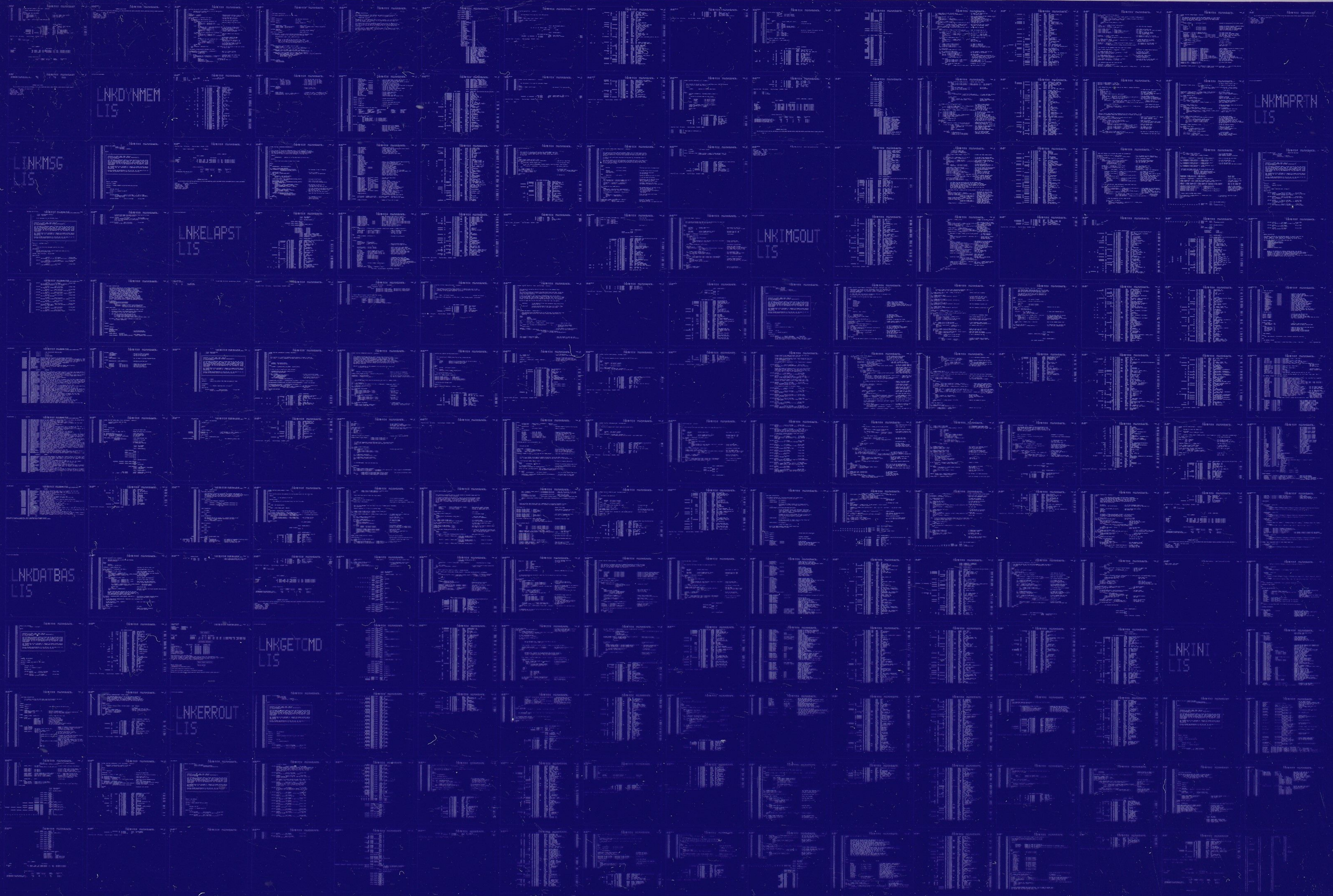
```
                                       50            6A  D0 000BC          MOVL    IMAGEHEADER, R0                         ; 2239
                                       51 00000000G  00  D0 000BF          MOVL    LNK$GL_FIXISD, R1                       :
                                                     0F  13 000C6          BEQL    5$                                      :
                                       51            18  C0 000C8          ADDL2   #24, R1                                 ; 2241
           51       04   A1            15            00  EF 000CB          EXTZV   #0, #21, 4(R1), R1                      ; 2242
                         51            51            09  78 000D1          ASHL    #9, R1, R1                              :
                                                     02  11 000D5          BRB     6$                                      :
                                                     51  D4 000D7 5$:      CLRL    R1                                      ; 2239
                                  2C   A0            51  D0 000D9 6$:      MOVL    R1, 44(R0)                              :
      20   A0            01            04       03   A8  F0 000DD          INSV    LNK$GL_CTLMSK+3, #4, #1, 32(R0)         ; 2246
                                       56       02   A0  3C 000E4          MOVZWL  2(R0), IHDACTIV                         ; 2247
                                       56            50  C0 000E8          ADDL2   R0, IHDACTIV                            :
                                       52            56  D0 000EB          MOVL    IHDACTIV, CURTFR                        ; 2248
                                          00000000G  00  D5 000EE          TSTL    LNK$GL_DBGTFPS                          ; 2250
                                                     0D  13 000F4          BEQL    7$                                      :
                                       62 00000000G  00  D0 000F6          MOVL    LNK$GL_DBGTFR, (CURTFR)                 ; 2252
                                       52       04   A6  9E 000FD          MOVAB   4(R6), CURTFR                           ; 2253
                                                     1B  11 00101          BRB     8$                                      ; 2250
                         17            68            06  E1 00103 7$:      BBC     #6, LNK$GL_CTLMSK, 8$                   ; 2255
                         7E 00000000G  00            14  C1 00107          ADDL3   #20, LNK$GL_IMGFIL, -(SP)               ; 2257
                                                     01  DD 0010F          PUSHL   #1                                      :
                                          00000000G  8F  DD 00111          PUSHL   #LIN$_DBGTFR                            :
                            00000000G  00            03  FB 00117          CALLS   #3, LIB$SIGNAL                          :
                                                10   A6  D4 0011E 8$:      CLRL    16(IHDACTIV)                            ; 2260
                         29            68            02  E1 00121          BBC     #2, LNK$GL_CTLMSK, 9$                   ; 2261
                                       50            6A  D0 00125          MOVL    IMAGEHEADER, R0                         ; 2263
                                  11   A0            02  90 00128          MOVB    #2, 17(R0)                              :
                                                08   AE  9F 0012C          PUSHAB  OTSEPNAM                                ; 2264
                                                     59  DD 0012F          PUSHL   R9                                      :
                                          00000000'  EF  9F 00131          PUSHAB  OTSINISHR                               :
                            00000000G  00            03  FB 00137          CALLS   #3, LNK$SEARCH                          :
                                       0D            50  E9 0013E          BLBC    R0, 9$                                  :
                                  10   A6       00   B9  D0 00141          MOVL    @LNK$GL_OTSEPDSC, 16(IHDACTIV)          ; 2266
                                       50            6A  D0 00146          MOVL    IMAGEHEADER, R0                         ; 2267
                                  20   A0       40   8F  88 00149          BISB2   #64, 32(R0)                             :
                         2B            68            03  E0 0014E 9$:      BBS     #3, LNK$GL_CTLMSK, 11$                  ; 2272
                                                08   AE  9F 00152          PUSHAB  OTSEPNAM                                ; 2273
                                                     59  DD 00155          PUSHL   R9                                      :
                                          00000000'  EF  9F 00157          PUSHAB  OTSENTRYSYM                             :
                            00000000G  00            03  FB 0015D          CALLS   #3, LNK$SEARCH                          :
                                       16            50  E9 00164          BLBC    R0, 11$                                 :
                                       62       00   B9  D0 00167          MOVL    @LNK$GL_OTSEPDSC, (CURTFR)              ; 2275
                                       50       04   A6  9E 0016B          MOVAB   4(R6), R0                               ; 2276
                                       50            52  D1 0016F          CMPL    CURTFR, R0                              :
                                                     06  12 00172          BNEQ    10$                                     :
                                       52       08   A6  9E 00174          MOVAB   8(R6), CURTFR                           ; 2277
                                                     03  11 00178          BRB     11$                                     :
                                       52            50  D0 0017A 10$:     MOVL    R0, CURTFR                              ; 2278
                                       50            6A  D0 0017D 11$:     MOVL    IMAGEHEADER, R0                         ; 2281
                                  20   A0            02  88 00180          BISB2   #2, 32(R0)                              :
                                          00000000G  00  D5 00184          TSTL    LNK$GL_TFRPSC                           ; 2283
                                                     28  13 0018A          BEQL    14$                                     :
                                  20   A0            02  8A 0018C          BICB2   #2, 32(R0)                              ; 2285
                         05            68            03  E1 00190          BBC     #3, LNK$GL_CTLMSK, 12$                  ; 2286
                         09       03   A8            01  E1 00194          BBC     #1, LNK$GL_CTLMSK+3, 13$                ; 2288
                                       62 00000000G  00  D0 00199 12$:     MOVL    LNK$GL_TFRADR, (CURTFR)                 ; 2290
                                                     2D  11 001A0          BRB     16$                                     :
```

```
                         7E 00000000G  00            14  C1 001A2 13$:    ADDL3   #20, LNK$GL_IMGFIL, -(SP)                    ; 2292
                                                     01  DD 001AA         PUSHL   #1                                           ;
                                          00000000G  8F  DD 001AC         PUSHL   #LIN$_TFRSYS                                 ;
                                                     14  11 001B2         BRB     15$                                          ;
                         17            68            01  E1 001B4 14$:    BBC     #1, LNK$GL_CTLMSK, 16$                       ; 2295
                         7E 00000000G  00            14  C1 001B8         ADDL3   #20, LNK$GL_IMGFIL, -(SP)                    ; 2297
                                                     01  DD 001C0         PUSHL   #1                                           ;
                                          00000000G  8F  DD 001C2         PUSHL   #LIN$_USRTFR                                 ;
                            00000000G  00            03  FB 001C8 15$:    CALLS   #3, LIB$SIGNAL                               ;
                                       57            6A  D0 001CF 16$:    MOVL    IMAGEHEADER, R7                              ; 2300
                                       56        04  A7  3C 001D2         MOVZWL  4(R7), SYMDBGDATA                            ;
                                       56            57  C0 001D6         ADDL2   R7, SYMDBGDATA                               ;
                                   08  A6        F2  A9  B0 001D9         MOVW    LNK$GW_DSTBLKS, 8(SYMDBGDATA)                ; 2302
                                                     04  13 001DE         BEQL    17$                                          ;
                                       66        EA  A9  3C 001E0         MOVZWL  LNK$GW_DSTVBN, (SYMDBGDATA)                  ; 2303
                                   0A  A6 00000000G  00  B0 001E4 17$:    MOVW    LNK$GW_GSTRECS, 10(SYMDBGDATA)               ; 2305
                                                     05  13 001EC         BEQL    18$                                          ;
                                   04  A6        F4  A9  3C 001EE         MOVZWL  LNK$GW_GSTVBN, 4(SYMDBGDATA)                 ; 2306
                                   10  A6 00000000G  00  D0 001F3 18$:    MOVL    LNK$GL_DMTBYTES, 16(SYMDBGDATA)              ; 2308
                                                     05  13 001FB         BEQL    19$                                          ;
                                   0C  A6        A4  AA  D0 001FD         MOVL    DMTVBN, 12(SYMDBGDATA)                       ; 2309
                                   0C  A7  35303230  8F  D0 00202 19$:    MOVL    #892351024, 12(R7)                           ; 2319
                                       56        06  A7  3C 0020A         MOVZWL  6(R7), IMGIDAREA                             ; 2322
                                       56            57  C0 0020E         ADDL2   R7, IMGIDAREA                                ;
                                       50            6B  9A 00211         MOVZBL  LNK$GT_IMGNAM, R0                            ; 2324
                            00000000G  8F            50  D1 00214         CMPL    R0, #LNK$K_MAX_FILENAME_LENGTH               ;
                                                     07  1B 0021B         BLEQU   20$                                          ;
                                       50 00000000G  8F  D0 0021D         MOVL    #LNK$K_MAX_FILENAME_LENGTH, R0               ;
                                       6B            50  90 00224 20$:    MOVB    R0, LNK$GT_IMGNAM                            ;
                                       50 00000000G  00  9A 00227         MOVZBL  LNK$GT_IMGID, R0                             ; 2325
                                       0F            50  91 0022E         CMPB    R0, #15                                      ;
                                                     03  1B 00231         BLEQU   21$                                          ;
                                       50            0F  D0 00233         MOVL    #15, R0                                      ;
                            00000000G  00            50  90 00236 21$:    MOVB    R0, LNK$GT_IMGID                             ;
                                       50            6B  9A 0023D         MOVZBL  LNK$GT_IMGNAM, R0                            ; 2327
                                                     50  D6 00240         INCL    R0                                           ;
                         66            6B            50  28 00242         MOVC3   R0, LNK$GT_IMGNAM, (IMGIDAREA)               ;
                                       50 00000000G  00  9A 00246         MOVZBL  LNK$GT_IMGID, R0                             ; 2328
                                                     50  D6 0024D         INCL    R0                                           ;
                    28   A6 00000000G  00            50  28 0024F         MOVC3   R0, LNK$GT_IMGID, 40(IMGIDAREA)              ;
                                   38  A6 00000000G  00  7D 00258         MOVQ    LNK$GQ_ALOSTIM, 56(IMGIDAREA)                ; 2330
                                   40  A6            05  90 00260         MOVB    #5, 64(IMGIDAREA)                            ; 2332
                                   41  A6 00000000G  00  3C 00264         MOVZWL  LNK$AW_VERSION, 65(IMGIDAREA)                ; 2333
                                   43  A6            2D  D0 0026C         MOVL    #45, 67(IMGIDAREA)                           ; 2334
                                   44  A6 00000000G  00  3C 00270         MOVZWL  LNK$AW_VERSION+2, 68(IMGIDAREA)              ; 2335
                                                     57  DD 00278         PUSHL   R7                                           ; 2340
                                                     01  DD 0027A         PUSHL   #1                                           ;
                                       7E        EC  A9  3C 0027C         MOVZWL  LNK$GW_HDRBLKS, -(SP)                        ;
                                 FCF8  CF            03  FB 00280         CALLS   #3, WRITESECTION                             ;
                                       07            50  E9 00285         BLBC    R0, 23$                                      ;
                            00000000V  EF            00  FB 00288 22$:    CALLS   #0, LNK$CLOSIMGFIL                           ; 2341
                                                         04 0028F 23$:    RET                                                  ; 2343

; Routine Size:  656 bytes,    Routine Base:  $CODE$ + 1406
; 1861          2344  1 global routine lnk$closimgfil : novalue =
; 1862          2345  2 begin
; 1863          2346  2 !
; 1864          2347  2 !        ROUTINE TO CLOSE THE IMAGE FILE
; 1865          2348  2 !
; 1866          2349  2 local    errorcode,
; 1867          2350  2          imagefab : block [fab$c_bln, byte];
; 1868          2351  2
; 1869          2352  2 bind
; 1870          2353  2          auxfnb = lnk$gl_imgfil [fdb$t_auxfnb] : block [nam$c_bln, byte]; ! AUXILIARY FILENAME BLOCK
; 1871          2354  2
; 1872          2355  2 if .lnk$gw_imgifi eql 0                                            ! IF NO IMAGE RETURN NOW
; 1873          2356  2 then return;
; 1874          2357  2
; 1875        P 2358  2 $fab_init (fab = imagefab                                        ! SET UP FAB TO CLOSE FILE
; 1876        P 2359  2           ,nam = auxfnb                                          ! SET POINTER TO NAM
; 1877        P 2360  2           ,fop = tef                                             ! SET TO TRUNCATE TO END OF FILE
; 1878          2361  2           );
; 1879          2362  2 imagefab [fab$w_ifi] = .lnk$gw_imgifi;                            ! JAM THE SAVED IFI
; 1880          2363  2
; 1881          2364  2 if not .lnk$gl_ctlmsk [lnk$v_image]                               ! IF IMAGE DISABLED
; 1882          2365  2 then imagefab [fab$v_dlt] = true;                                !  THEN DELETE THE FILE
; 1883          2366  2
; 1884          2367  3 if not (errorcode = $close (fab=imagefab))                       ! DO IT AND IF AN ERROR
; 1885          2368  2 then signal ( lin$_closeout, 1
; 1886          2369  2             , lnk$gl_imgfil [fdb$q_filename]
; 1887          2370  2             , .errorcode, imagefab [fab$l_stv]
; 1888          2371  2             );
; 1889          2372  2 lnk$gw_imgifi = 0;                                                ! CLEAR THE ID
; 1890          2373  2
; 1891          2374  2 return true                                                       ! AND ALL DONE
; 1892          2375  1 end;



                                    03FC 00000          .ENTRY  LNK$CLOSIMGFIL, Save R2,R3,R4,R5,R6,R7,R8,- ; 2344
                                                                R9                                          ;
                  59 00000000G   00   9E 00002          MOVAB   LNK$GL_IMGFIL, R9                           ;
                  58 00000000'   EF   9E 00009          MOVAB   LNK$GW_IMGIFI, R8                           ;
                  5E         B0  AE   9E 00010          MOVAB   -80(SP), SP                                 ;
            56    69             26   C1 00014          ADDL3   #38, LNK$GL_IMGFIL, R6                      ; 2353
                  57             68   3C 00018          MOVZWL  LNK$GW_IMGIFI, R7                           ; 2355
                                 57   13 0001B          BEQL    3$                                          ;
0050  8F    00    6E             00   2C 0001D          MOVC5   #0, (SP), #0, #80, $RMS_PTR                 ; 2361
                                 6E      00024                                                              ;
                  6E       5003  8F   B0 00025          MOVW    #20483, $RMS_PTR                            ;
             04   AE   10000000  8F   D0 0002A          MOVL    #268435456, $RMS_PTR+4                      ;
             16   AE             02   90 00032          MOVB    #2, $RMS_PTR+22                             ;
             1F   AE             02   90 00036          MOVB    #2, $RMS_PTR+31                             ;
             28   AE             56   D0 0003A          MOVL    R6, $RMS_PTR+40                             ;
             02   AE             57   B0 0003E          MOVW    R7, IMAGEFAB+2                              ; 2362
                  05 00000000G   00   E8 00042          BLBS    LNK$GL_CTLMSK, 1$                           ; 2364
             05   AE         80  8F   88 00049          BISB2   #128, IMAGEFAB+5                            ; 2365
                                 5E   DD 0004E 1$:      PUSHL   SP                                          ; 2367
        00000000G 00             01   FB 00050          CALLS   #1, SYS$CLOSE                               ;
```

```
                                        18             50 E8 00057          BLBS    ERRORCODE, 2$                                   ;
                                                  0C   AE 9F 0005A          PUSHAB  IMAGEFAB+12                                     ; 2370
                                                       50 DD 0005D          PUSHL   ERRORCODE                                       ;
                          7E            69             14 C1 0005F          ADDL3   #20, LNK$GL_IMGFIL, -(SP)                       ; 2369
                                                       01 DD 00063          PUSHL   #1                                              ;
                                            00000000G  8F DD 00065          PUSHL   #LIN$_CLOSEOUT                                  ;
                             00000000G  00             05 FB 0006B          CALLS   #5, LIB$SIGNAL                                  ;
                                                       68 B4 00072 2$:      CLRW    LNK$GW_IMGIFI                                   ; 2372
                                                          04 00074 3$:      RET                                                     ; 2375

; Routine Size:  117 bytes,    Routine Base:  $CODE$ + 1696


; 1893          2376  1
; 1894          2377  0 end eludom



                                                                    .EXTRN  LIB$SIGNAL

;                                           PSECT SUMMARY
;
;       Name                          Bytes                        Attributes
;
;    $GLOBAL$                           128  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;    $OWN$                              128  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;    $PLIT$                             107  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;    $CODE$                            5899  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;    .  ABS  .                            0  NOVEC,NOWRT,NORD ,NOEXE,NOSHR,  LCL,  ABS,  CON,NOPIC,ALIGN(0)




;                                    Library Statistics
;
;                                                  -------- Symbols --------       Pages        Processing
;       File                                       Total     Loaded   Percent       Mapped       Time
;
;    _$255$DUA28:[SYSLIB]LIB.L32;1                 18619        187         1       1000           00:01.9
;    _$255$DUA28:[LINKER.OBJ]DATBAS.L32;1            538         62        11         28           00:00.8





;                                           COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:LNKIMGOUT/OBJ=OBJ$:LNKIMGOUT MSRC$:LNKIMGOUT/UPDATE=(ENH$:LNKIMGOUT)

; Size:           5899 code + 363 data bytes
; Run Time:          01:56.4
; Elapsed Time:      03:53.2
; Lines/CPU Min:      1225
; Lexemes/CPU-Min: 25962
; Memory Used:  630 pages
; Compilation Complete
```

0216 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



LNKDYNMEM
LIS

LNKMAPRTN
LIS

LINKMSG
LIS

LNKELAPST
LIS

LNKIMGOUT
LIS

LNKDATBAS
LIS

LNKGETCMD
LIS

LNKINI
LIS

LNKERROUT
LIS